Transfer Learning and Applications in Computational Biology

Gunnar Rätsch[1], **Christian Widmer**[1,2], **Marius Kloft**[1,3], **Nico Görnitz**[2], **Gabriele Schweikert**

[1] Memorial Sloan-Kettering Cancer Center, NY, USA

[2] Technical University of Berlin, Germany

[3] New York University, NY, USA

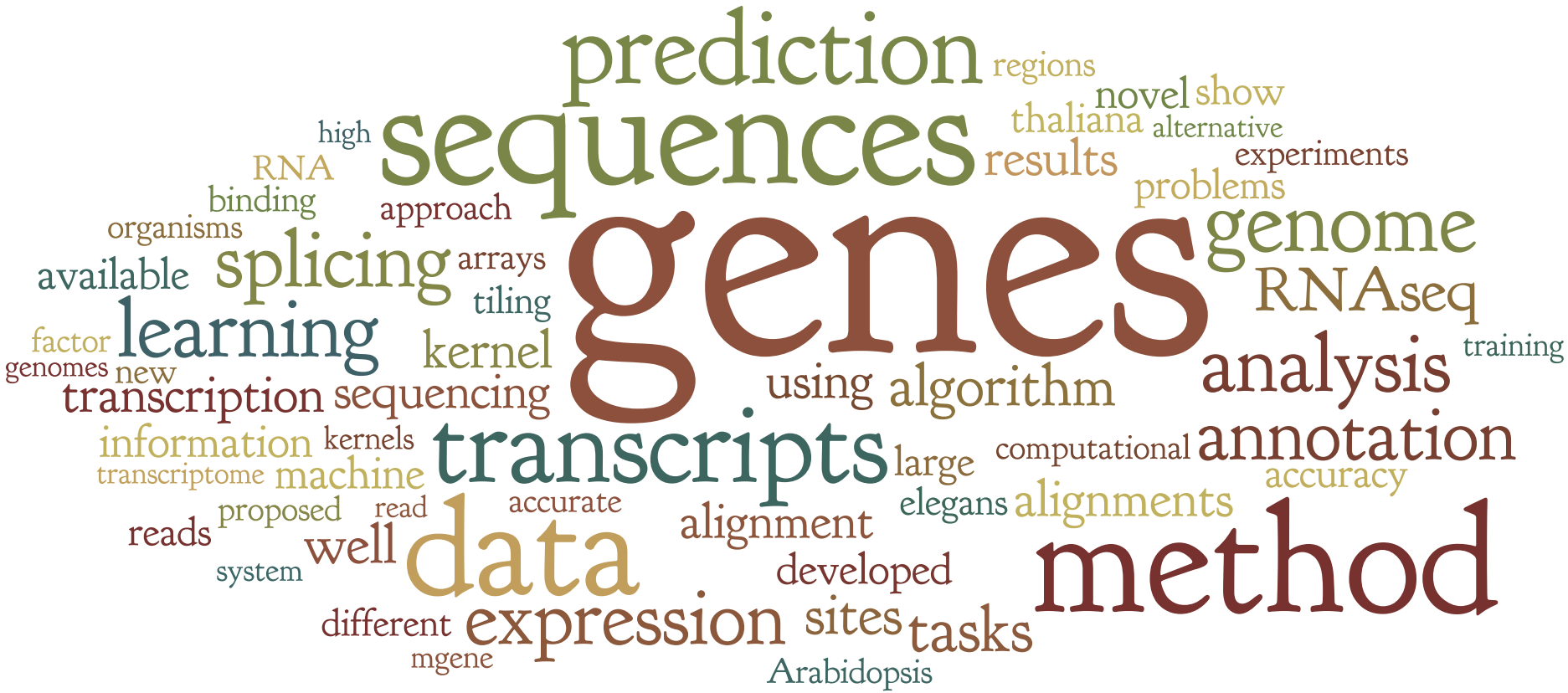
Memorial Sloan-Kettering Cancer Center

cBio@MSKCC



Frequent words of abstracts from publications 1998-2004.

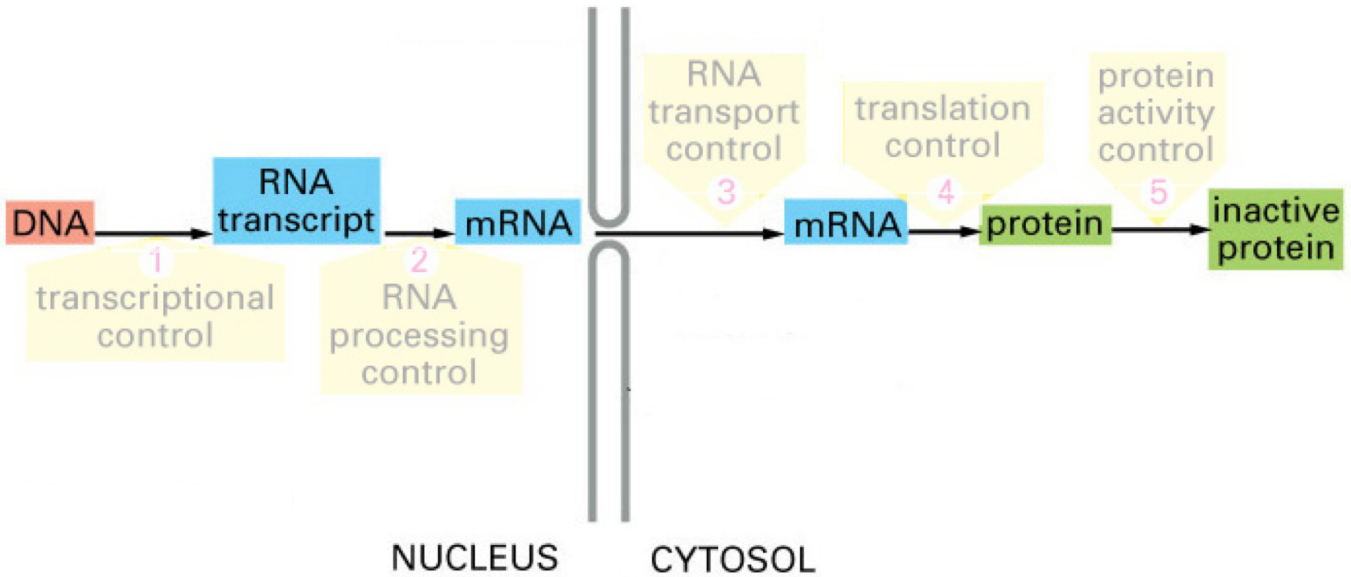
[wordle.net]



Frequent words of abstracts from publications 2005-2012.

[wordle.net]

Learning About the Central Dogma of Biology



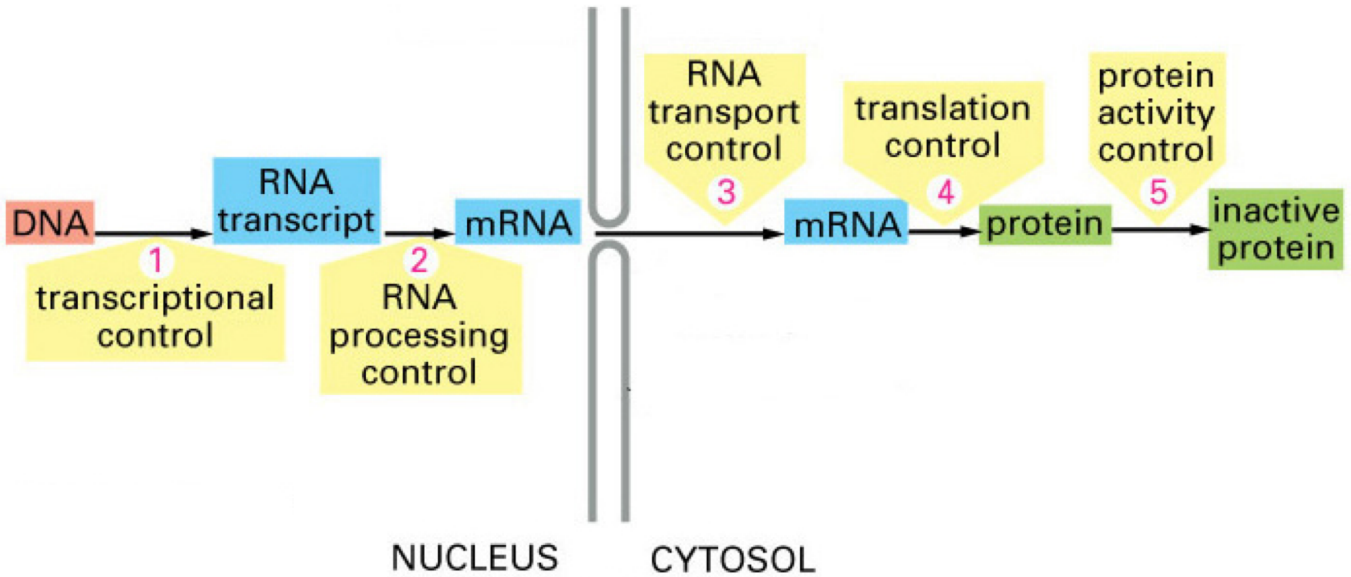
Goal: Learn to predict what these processes accomplish:

- Given the DNA, . . . , predict all gene products

$$f(\text{DNA}, \quad) = \text{RNA} \qquad g(\text{RNA}, \quad) = \text{protein}$$

- Estimating f, g amounts to cracking the codes of transcription, epigenetics, splicing, . . .

Learning About the Central Dogma of Biology



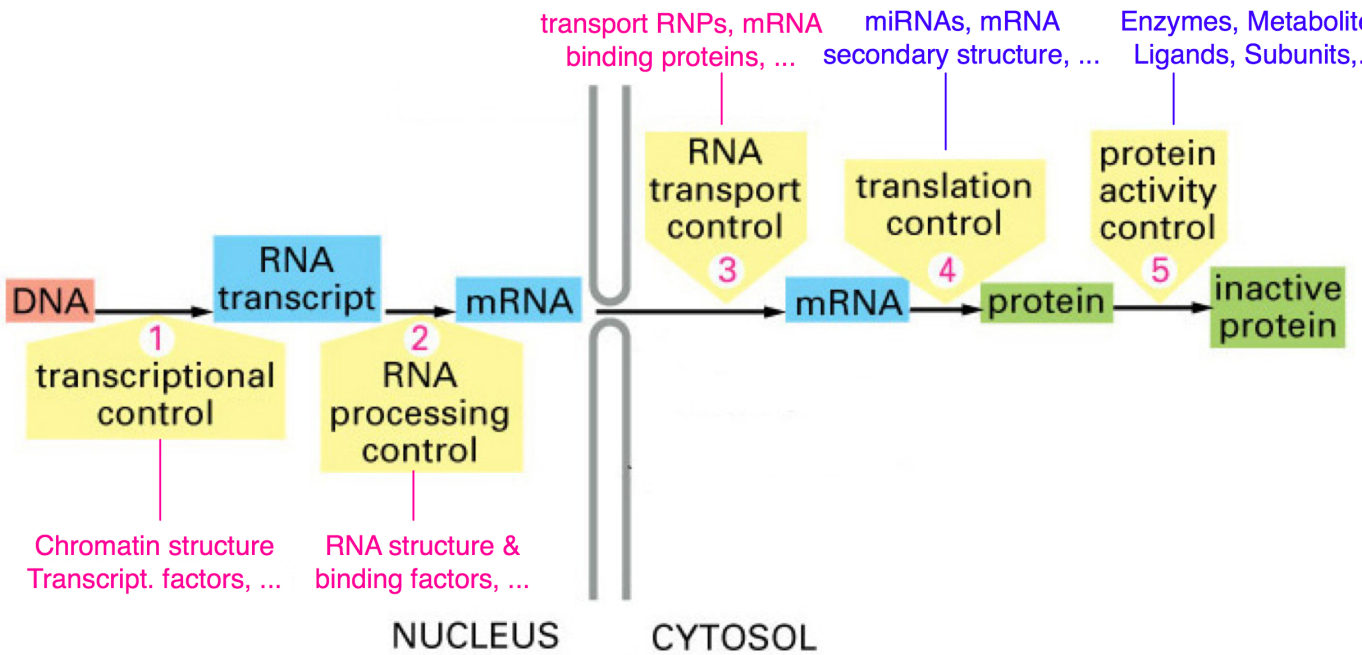
Goal: Learn to predict what these processes accomplish:

- Given the DNA, . . . , predict all gene products

  $f(\text{DNA}, \boxed{1}\,\boxed{2}\,\boxed{3}) = \text{RNA}$ $\qquad$ $g(\text{RNA}, \boxed{4}\,\boxed{5}) = \text{protein}$

- Estimating f, g amounts to cracking the codes of transcription, epigenetics, splicing, . . .

Learning About the Central Dogma of Biology



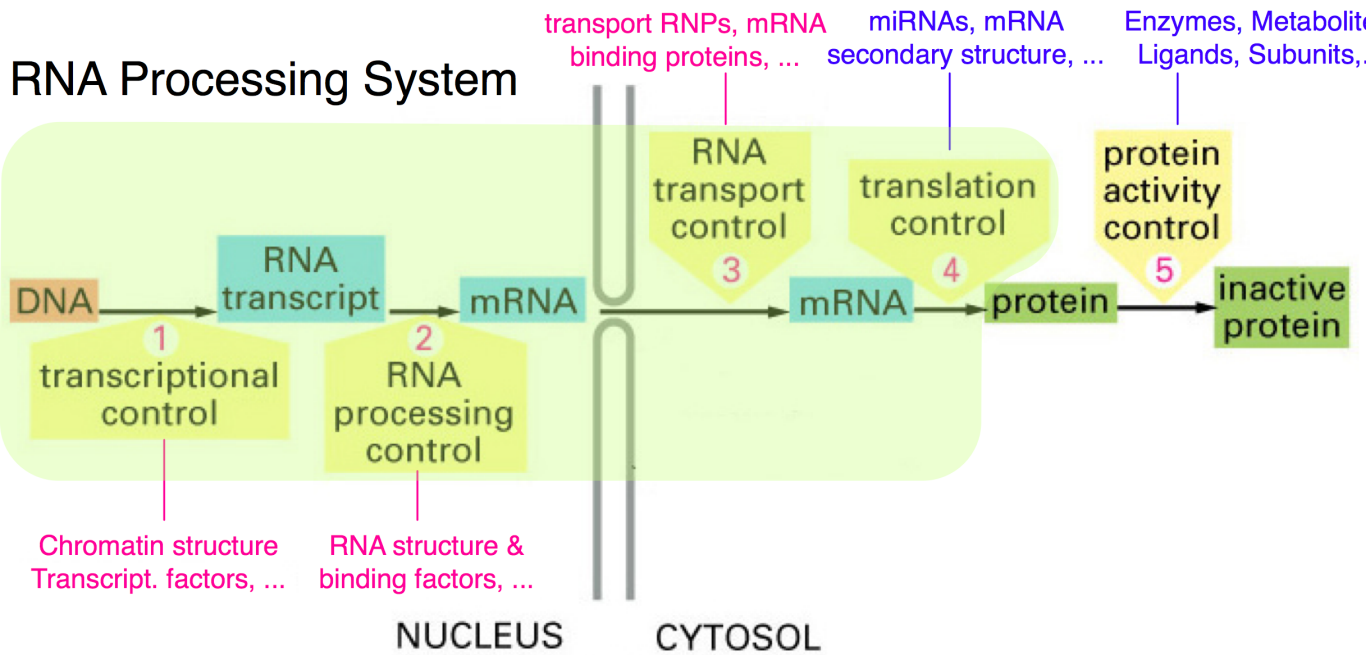
Goal: Learn to predict what these processes accomplish:

- Given the DNA, ..., predict all gene products

$$f(\text{DNA}, \boxed{1}\,\boxed{2}\,\boxed{3}) = \text{RNA} \qquad g(\text{RNA}, \boxed{4}\,\boxed{5}) = \text{protein}$$

- Estimating f, g amounts to cracking the codes of transcription, epigenetics, splicing, ...

Learning About the Central Dogma of Biology



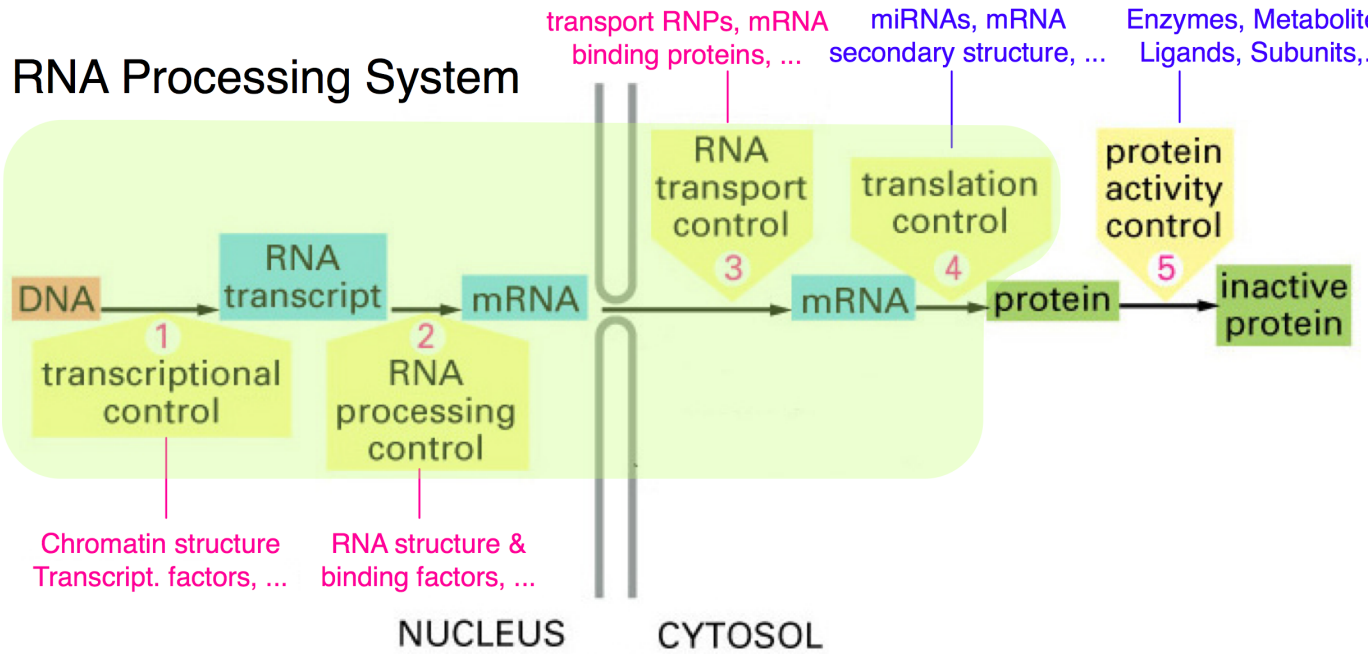
Goal: Learn to predict what these processes accomplish:

- Given the DNA, . . . , predict all gene products

$$f(\text{DNA}, \boxed{1}\boxed{2}\boxed{3}) = \text{RNA} \qquad g(\text{RNA}, \boxed{4}\boxed{5}) = \text{protein}$$

- Estimating f, g amounts to cracking the codes of transcription, epigenetics, splicing, . . .

Learning About the Central Dogma of Biology



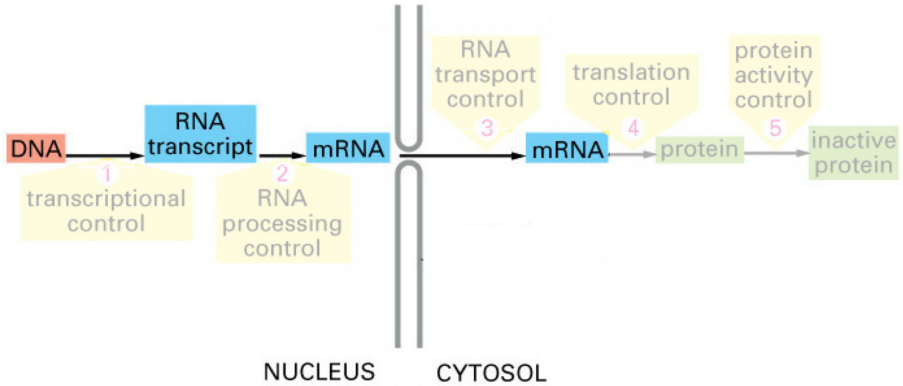
Three things will be crucial:

- Biological insights
- Many observations of the system: $(\text{DNA}, \boxed{1\ 2\ 3}, \text{RNA})_{i=1}^{N}$
- Empirical inference to estimate Θ: $f_\Theta(\text{DNA}, \boxed{1\ 2\ 3}) = \text{RNA}$

Learning About the Central Dogma

Goal: Estimate f to predict RNAs

✔ ✔ ? ?
$f(\text{DNA}, \boxed{1\ 2\ 3}) = \text{RNA}$



Need:

- Good inference method
- Inputs (DNA, 1 2 3)
- Outputs (complete transcriptome)

✔ ?
?
?

Challenges:

1. RNA only partially known
2. Factors 1 2 3 only partially known
3. Improved inference methods

Recent Machine Learning Work

⇝ Develop fast, accurate and interpretable learning methods

1. Large scale sequence classification

   [Rätsch et al., 2006a; Sonnenburg et al., 2010, 2007; Toussaint et al., 2010]

2. Analysis and explanation of learning results

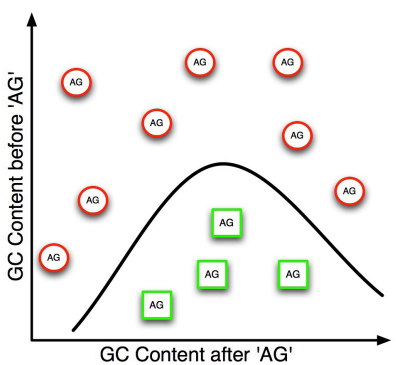
   [Rätsch et al., 2006b; Sonnenburg et al., 2008; Zien et al., 2009]

3. Sequence segmentation & structure prediction

   [Rätsch et al., 2007; Schweikert et al., 2009; Zeller et al., 2008]

4. Transfer & Multitask learning

   [Schweikert et al., 2008a; Widmer et al., 2011, 2012, 2010a; Widmer and Rätsch, 2011; Widmer et al., 2010c]



Recent Machine Learning Work

⇝ Develop fast, accurate and interpretable learning methods

1. Large scale sequence classification

   [Rätsch et al., 2006a; Sonnenburg et al., 2010, 2007; Toussaint et al., 2010]

2. Analysis and explanation of learning results

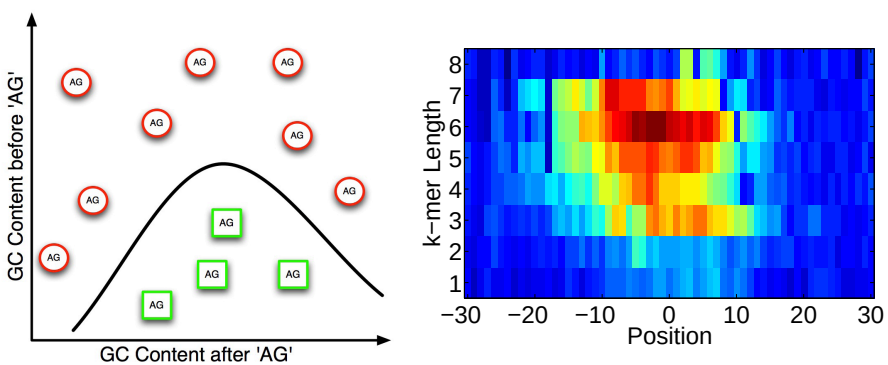
   [Rätsch et al., 2006b; Sonnenburg et al., 2008; Zien et al., 2009]

3. Sequence segmentation & structure prediction

   [Rätsch et al., 2007; Schweikert et al., 2009; Zeller et al., 2008]

4. Transfer & Multitask learning

   [Schweikert et al., 2008a; Widmer et al., 2011, 2012, 2010a; Widmer and Rätsch, 2011; Widmer et al., 2010c]



Recent Machine Learning Work

⇝ Develop fast, accurate and interpretable learning methods

1. Large scale sequence classification

   [Rätsch et al., 2006a; Sonnenburg et al., 2010, 2007; Toussaint et al., 2010]

2. Analysis and explanation of learning results

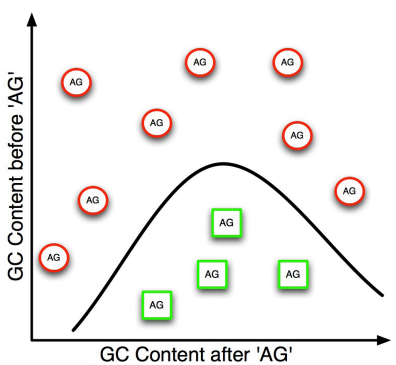
   [Rätsch et al., 2006b; Sonnenburg et al., 2008; Zien et al., 2009]

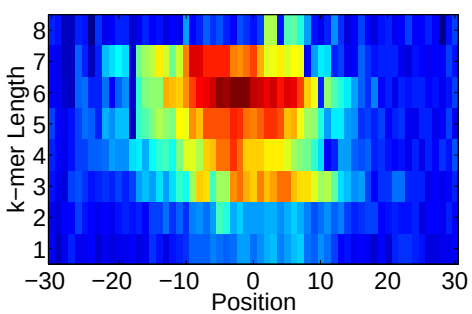
3. Sequence segmentation & structure prediction

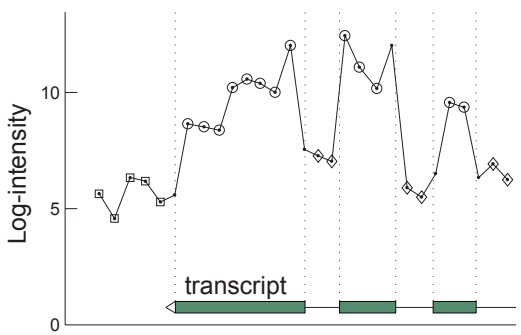
   [Rätsch et al., 2007; Schweikert et al., 2009; Zeller et al., 2008]

4. Transfer & Multitask learning

   [Schweikert et al., 2008a; Widmer et al., 2011, 2012, 2010a; Widmer and Rätsch, 2011; Widmer et al., 2010c]







Recent Machine Learning Work

⇝ Develop fast, accurate and interpretable learning methods

1. Large scale sequence classification

   [Rätsch et al., 2006a; Sonnenburg et al., 2010, 2007; Toussaint et al., 2010]

2. Analysis and explanation of learning results

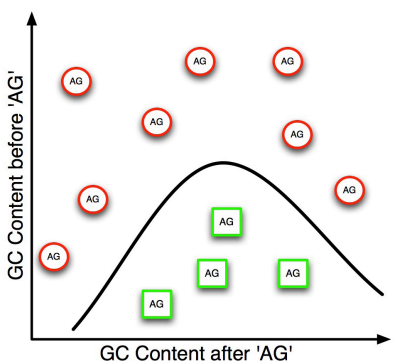
   [Rätsch et al., 2006b; Sonnenburg et al., 2008; Zien et al., 2009]

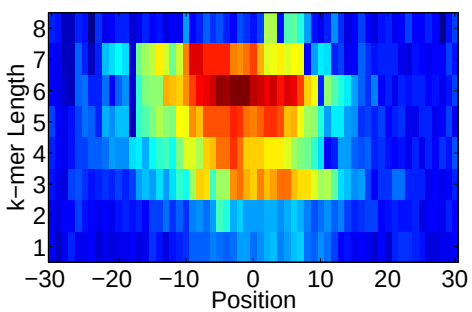
3. Sequence segmentation & structure prediction

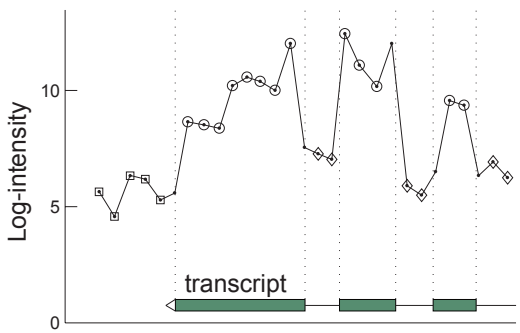
   [Rätsch et al., 2007; Schweikert et al., 2009; Zeller et al., 2008]

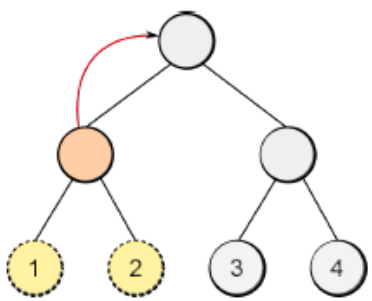
4. Transfer & Multitask learning

   [Schweikert et al., 2008a; Widmer et al., 2011, 2012, 2010a; Widmer and Rätsch, 2011; Widmer et al., 2010c]







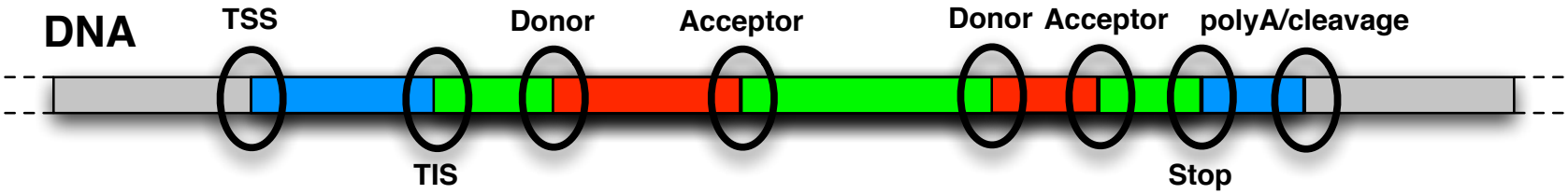




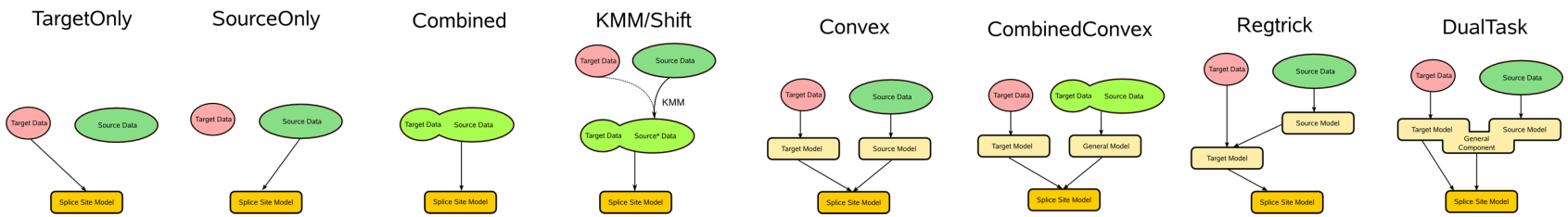
Many algorithms implemented in Shogun toolbox (GPL, $\geq$ 1000 users)

Roadmap

- Motivation from computational biology



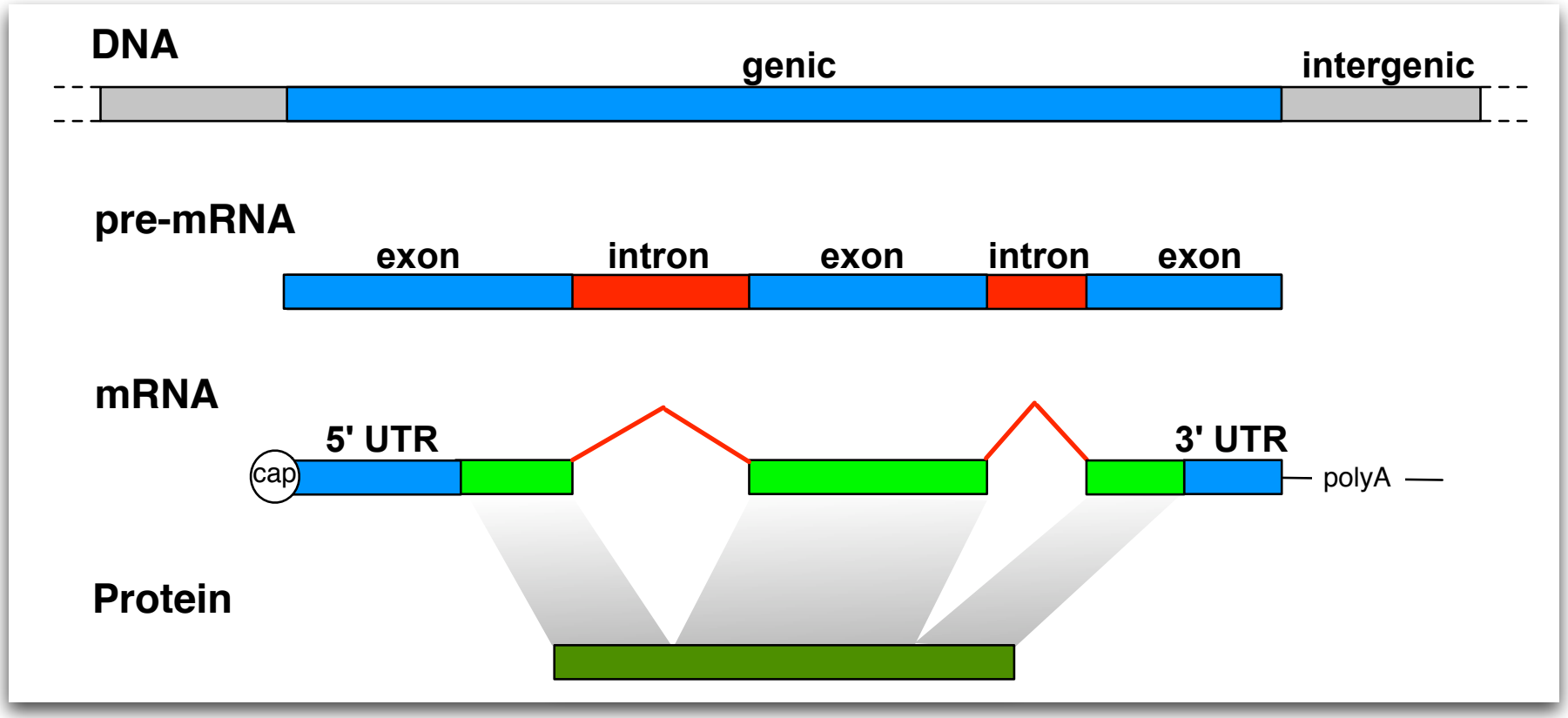
- Empirical comparison of domain adaptation algorithms



- Algorithms for hierarchical multi-task learning
- Algorithms for learning task relations
- Fast(er) Algorithms
- Discussion & Conclusion

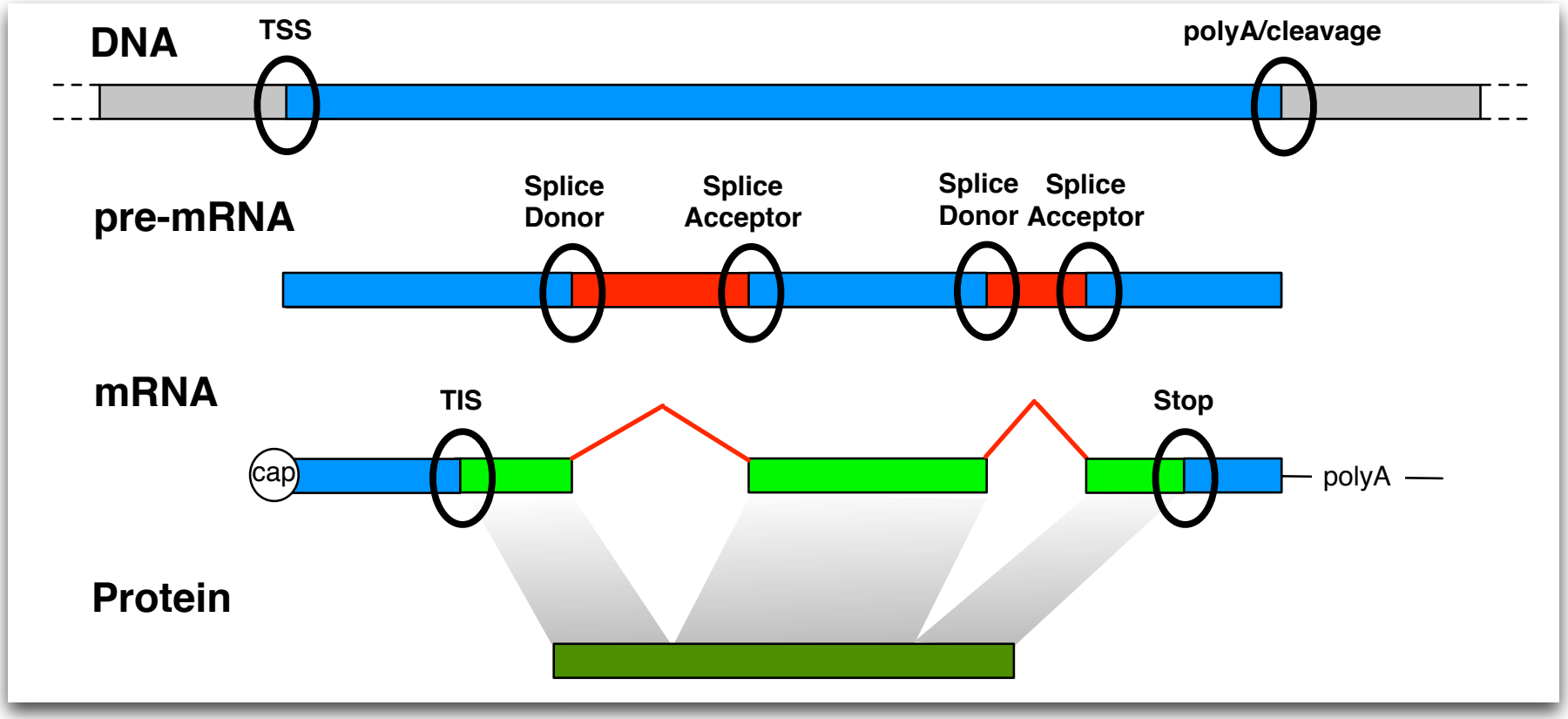


A Core CompBio Problem: Gene Finding



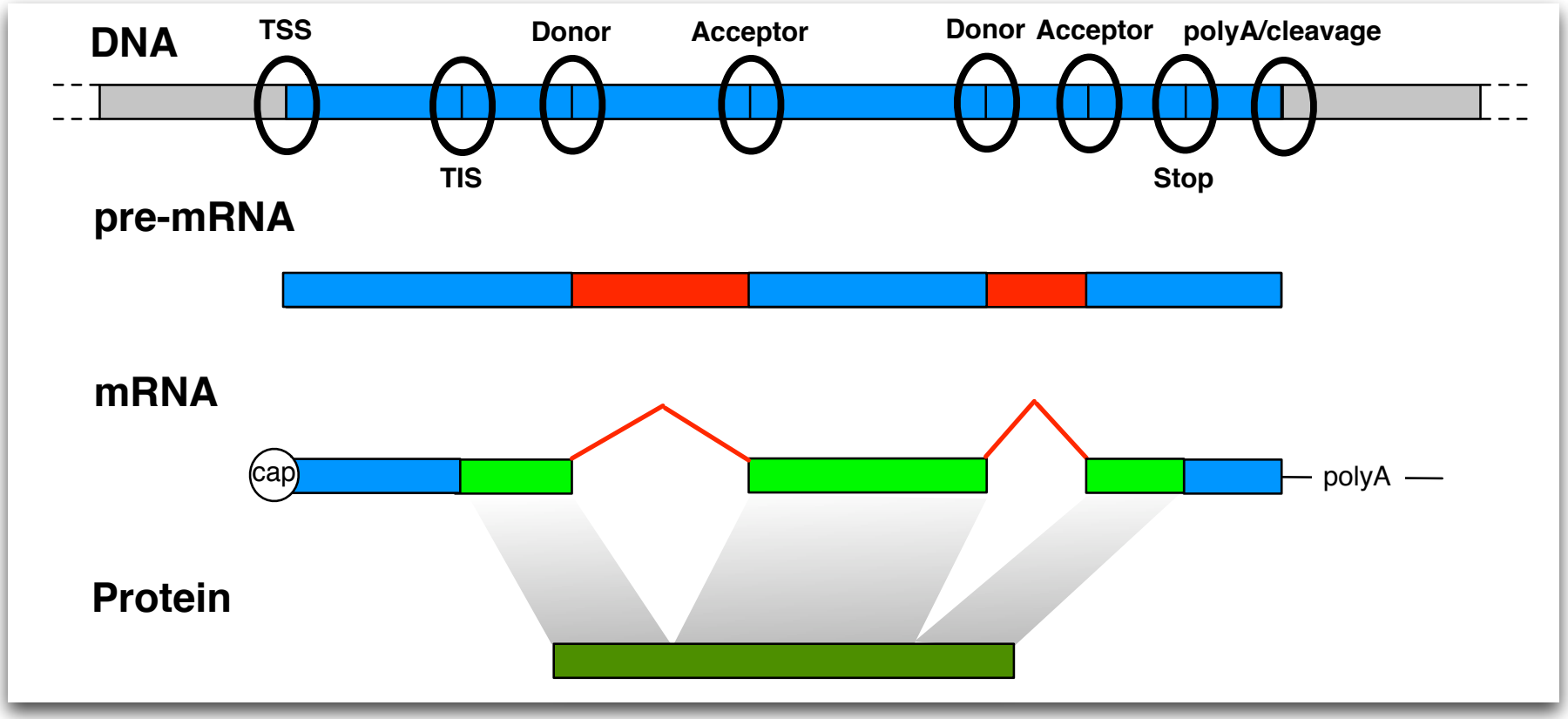
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

A Core CompBio Problem: Gene Finding



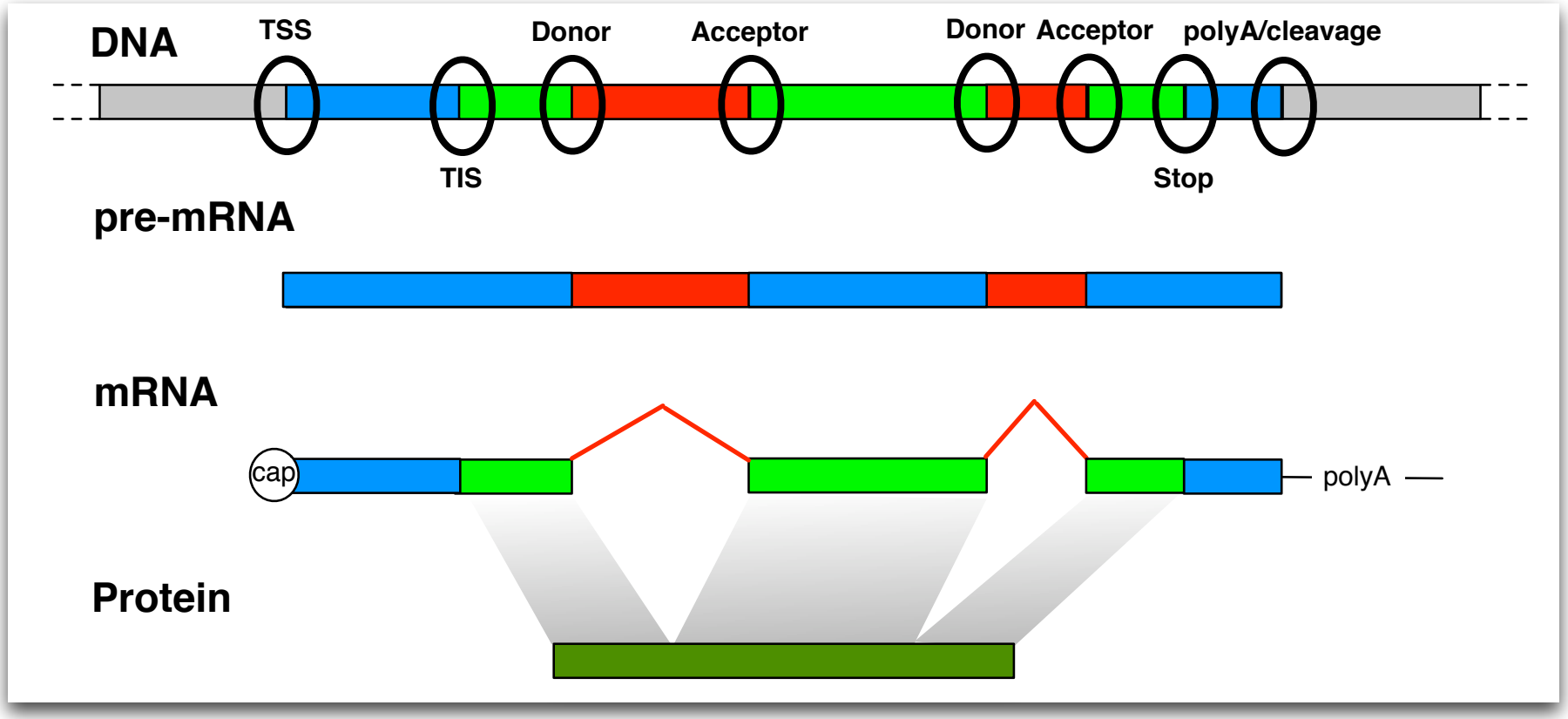
- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

A Core CompBio Problem: Gene Finding



- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

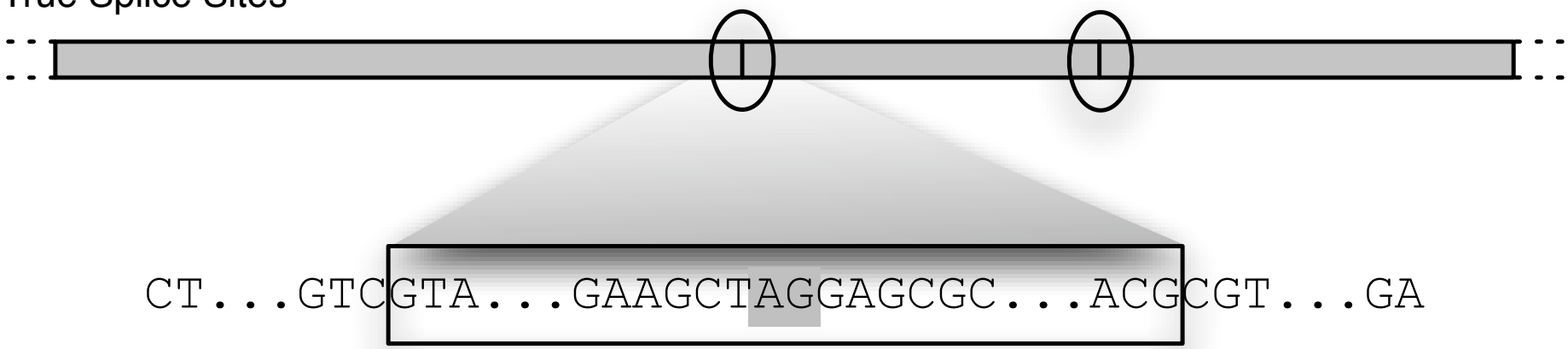
A Core CompBio Problem: Gene Finding



- Given a piece of DNA sequence
- Predict gene products including intermediate processing steps
- Predict signals used during processing
- Predict the correct corresponding **label sequence** with labels

Example: Splice Site Recognition

True Splice Sites



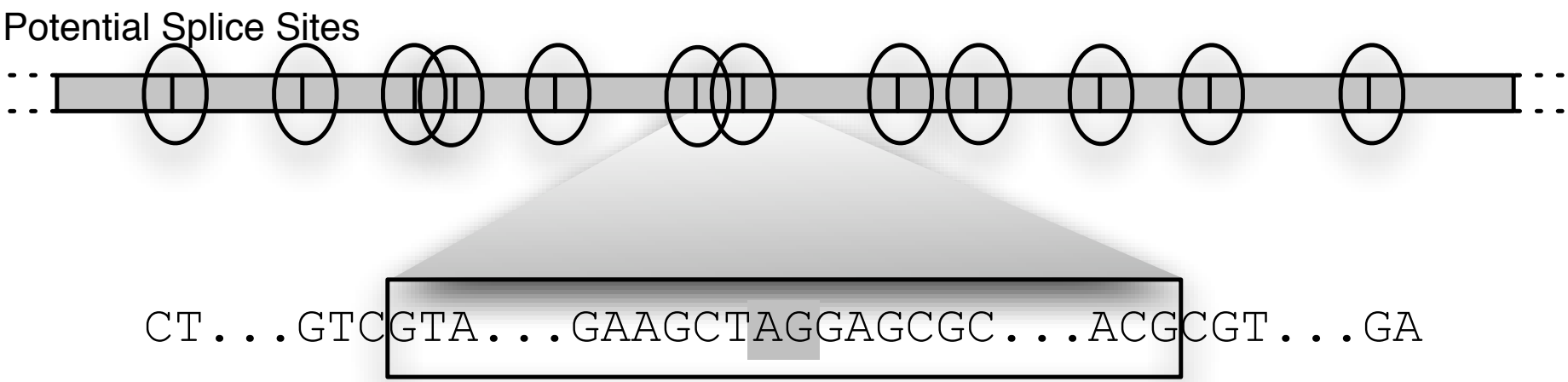
CT...GTCGTA...GAAGCTAGGAGCGC...ACGCGT...GA

≈ 150 nucleotides window around dimer



```
1  GCCAATATTTTTCTATTCAGGTGCAATCAATCACCCATCAT
1  ATTGAATGAACATATTCCAGGGTCTCCTTCCACCTCAACAA
1  AGCAACGAACTCCATTACAGCAAGGACATCGAAGTCGATCA
1  GCCAATTTTTGACCTTGCAGAATCAATCGTGCACGTTCGGA
```

Example: Splice Site Recognition



≈ 150 nucleotides window around dimer

1	GCCAATATTTTTCTATTCAGGTGCAATCAATCACCCATCAT
1	ATTGAATGAACATATTCCAGGGTCTCCTTCCACCTCAACAA
1	AGCAACGAACTCCATTACAGCAAGGACATCGAAGTCGATCA
1	GCCAATTTTTGACCTTGCAGAATCAATCGTGCACGTTCGGA
-1	CATCTGAAATTTCCCCCAAGTATAGCGGAAATAGACCGACG
-1	GAAATTTCCCCCAAGTATAGCGGAAATAGACCGACGAAATC
-1	CCCAAGTATAGCGGAAATAGACCGACGAAATCGCTCTCTCC
-1	AATCGCTCTCTCCCTGGGAGCGATGCGAATGTCAAATTCGA
-1	ACCAAAAAATCAATTTTTAGATTTTTCGAATTAATTTTTCG
-1	TGCTTTGCATGTTTCTAAAGTTACAGCCGTTCAAAATTTAA
-1	GCATGTTTCTAAAGTTACAGCCGTTCAAAATTTAAAAACTC
-1	ACCAATACGCAATGACTGAGTCTGTAATTTCACATAGTAAT

Domain Adaptation for Genome Annotation

Motivation:

- Increasing number of sequenced genomes
- Often newly sequenced genomes are poorly annotated
- However often relatives with good annotation exist
- **Idea:** Transfer knowledge between organisms

Example: Splice site annotation in worm genomes

- Newly sequenced organism: *C. briggsae*
  - $\approx$ 100 confirmed genes (590 splice site pairs)
- Well annotated relative: *C. elegans*
  - $\approx$ 10.000 confirmed genens (36.782 splice site pairs)

Domain Adaptation for Genome Annotation

Motivation:

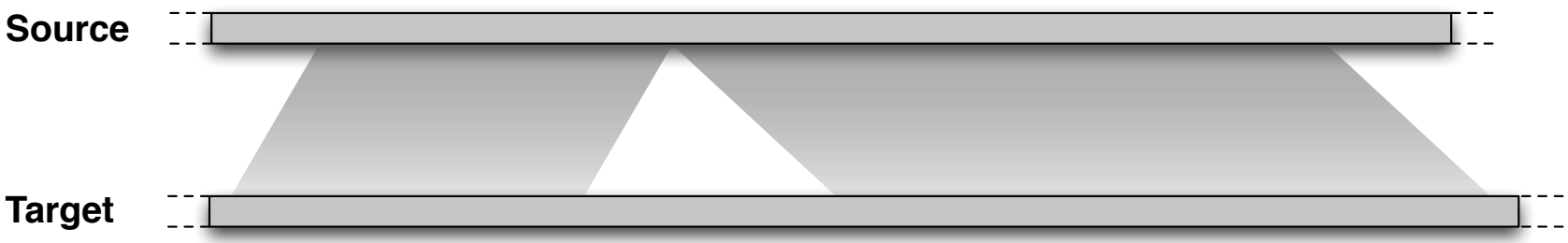
- Increasing number of sequenced genomes
- Often newly sequenced genomes are poorly annotated
- However often relatives with good annotation exist
- **Idea:** Transfer knowledge between organisms

Example: Splice site annotation in worm genomes

- Newly sequenced organism: *C. briggsae*
  - $\approx$ 100 confirmed genes (590 splice site pairs)
- Well annotated relative: *C. elegans*
  - $\approx$ 10.000 confirmed genens (36.782 splice site pairs)

The "Bioinformatics Way" of Transfer Learning

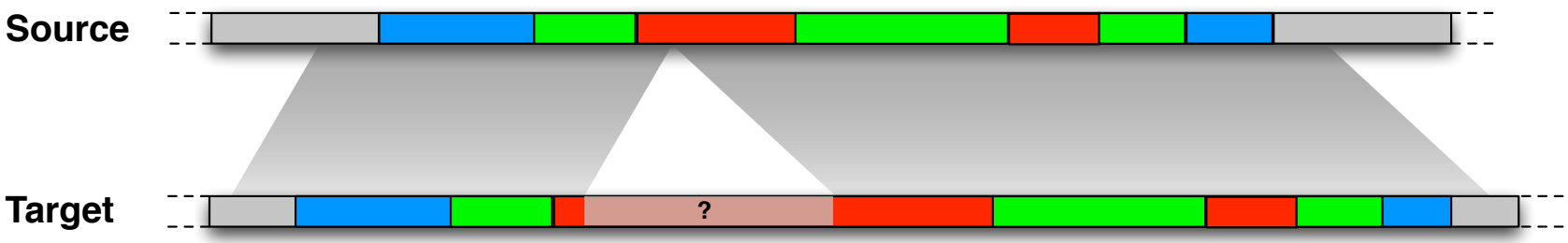
1. Homology-based annotation (a.k.a. "Comparative genomics")



Works for closely related species, does not require any labeled data from target organism.

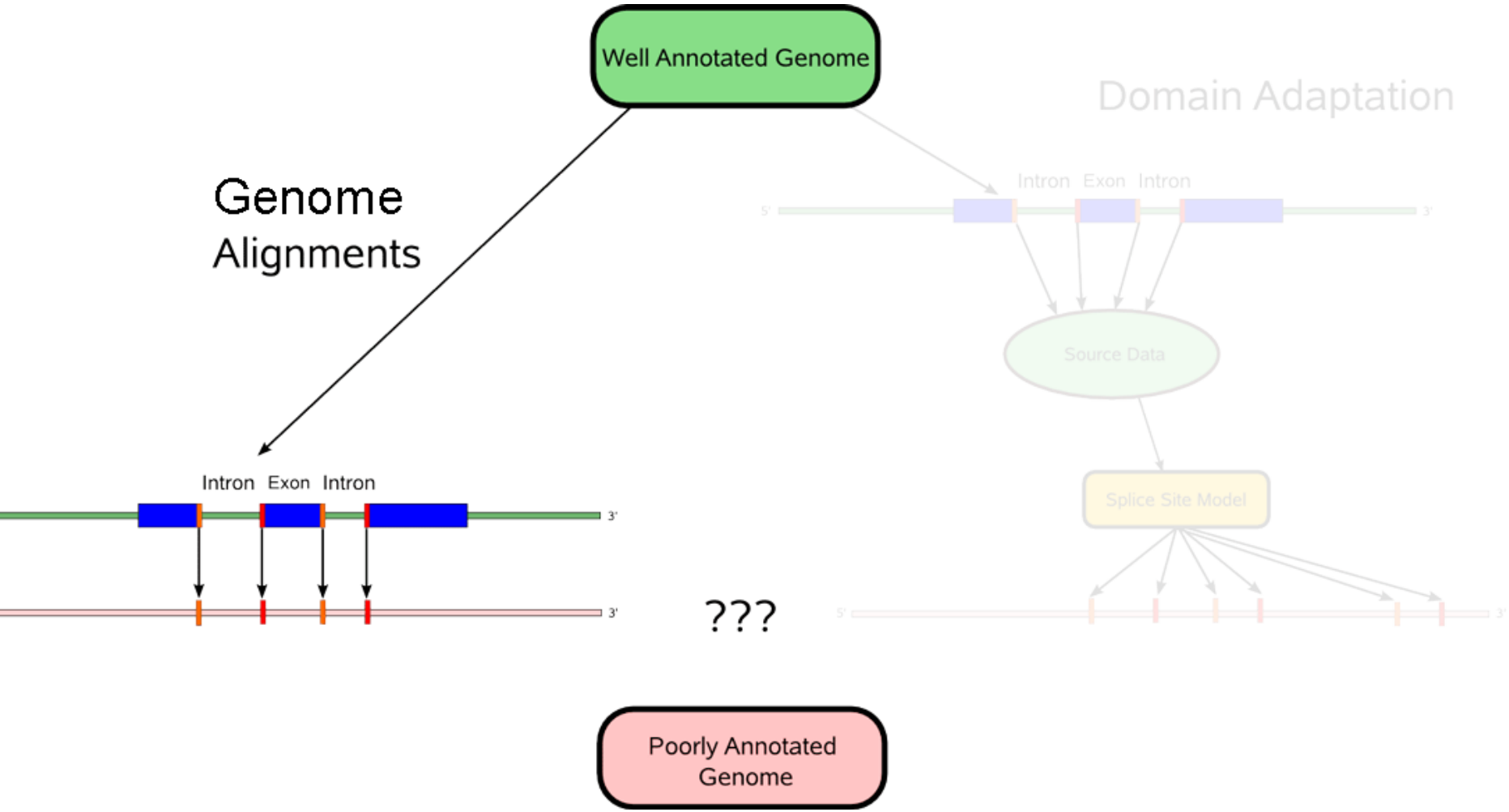
The "Bioinformatics Way" of Transfer Learning

1. Homology-based annotation (a.k.a. "Comparative genomics")



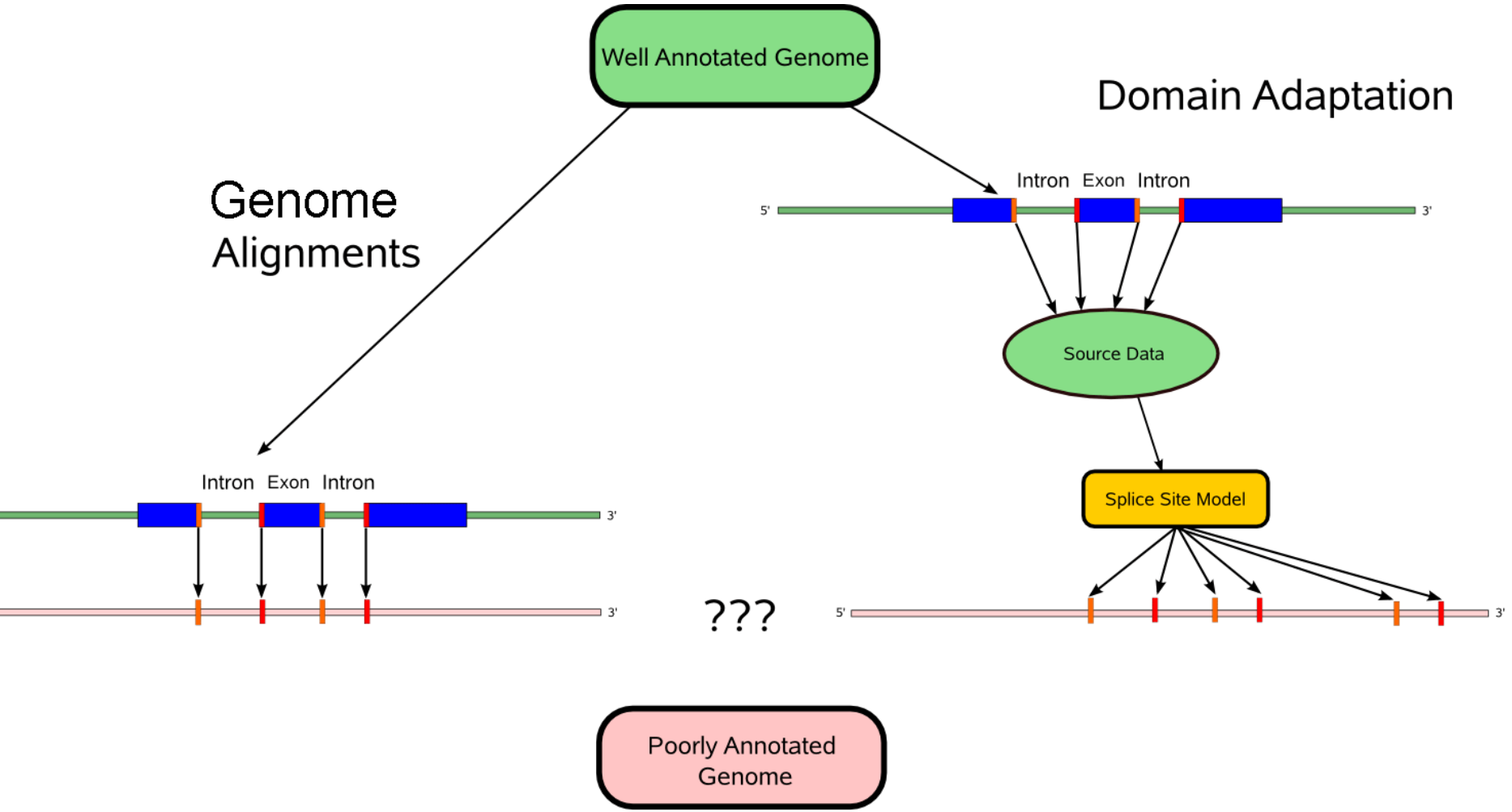
Works for closely related species, does not require any labeled data from target organism.

Domain Adaptation by Learning vs. Homology



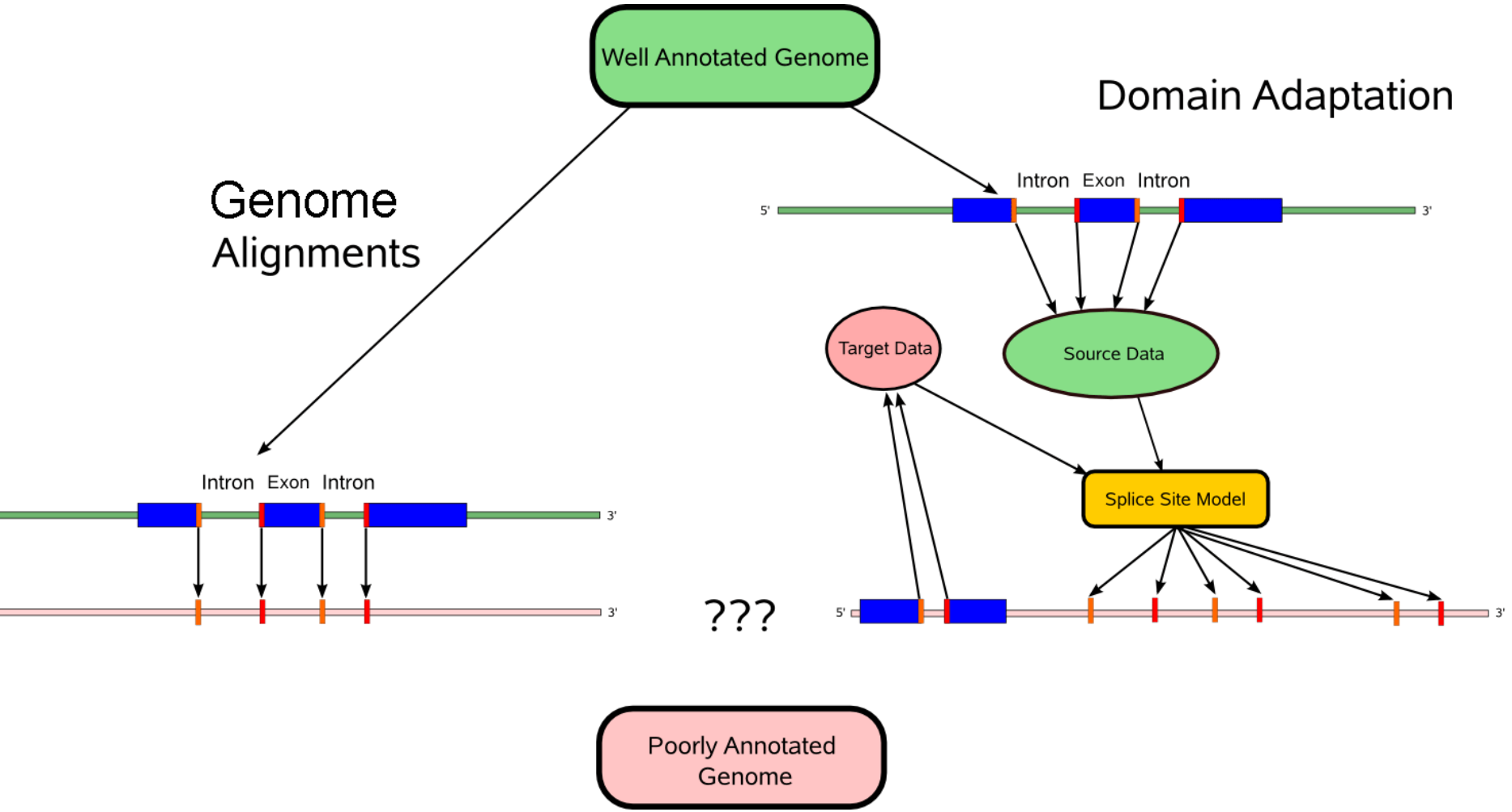
[Schweikert et al., 2008b; Widmer et al., 2010c]

Domain Adaptation by Learning vs. Homology



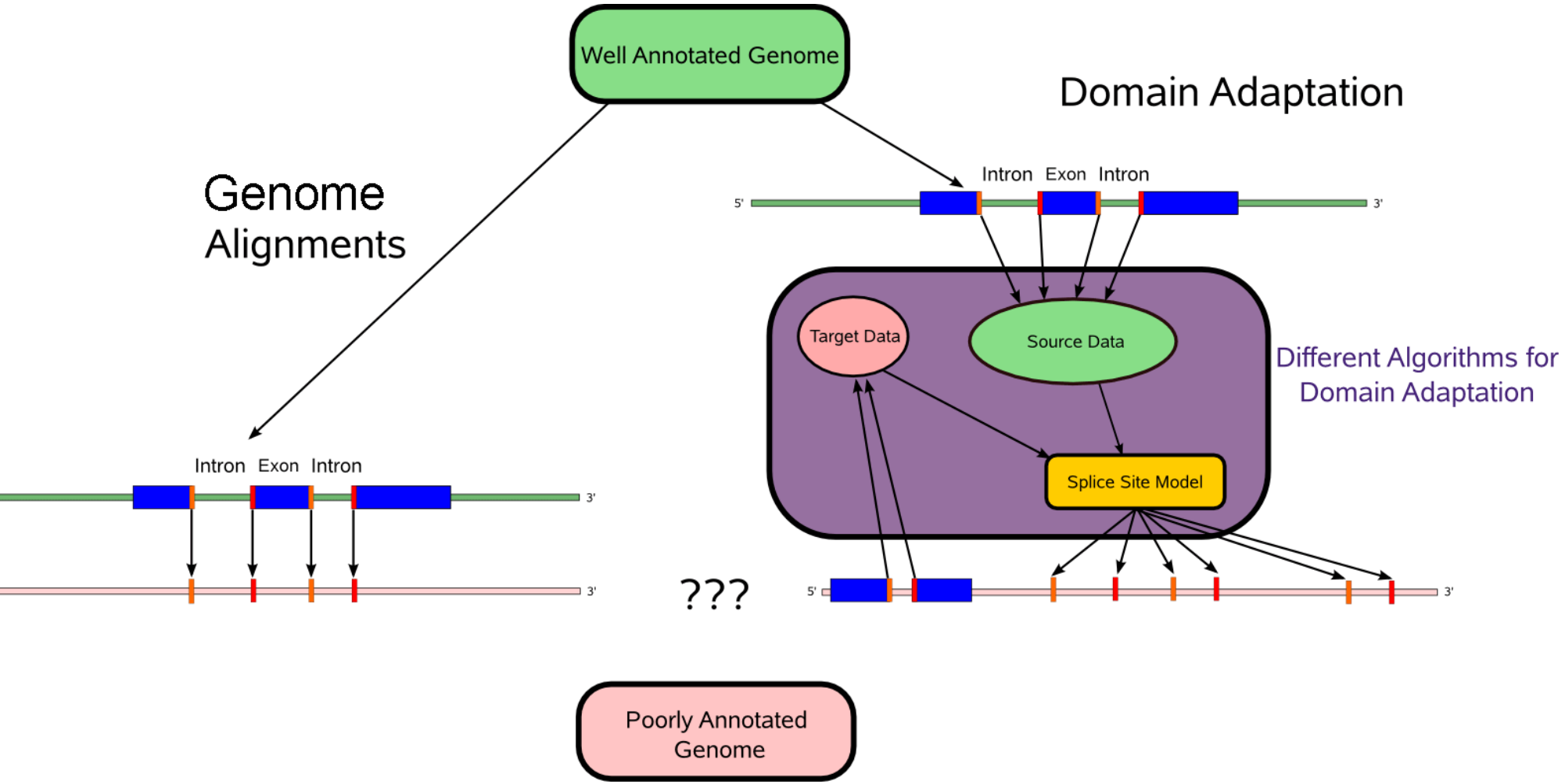
[Schweikert et al., 2008b; Widmer et al., 2010c]

Domain Adaptation by Learning vs. Homology



[Schweikert et al., 2008b; Widmer et al., 2010c]

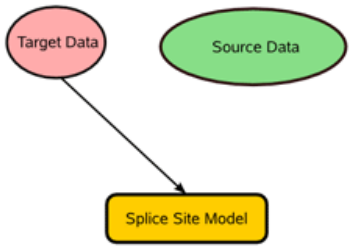
Domain Adaptation by Learning vs. Homology



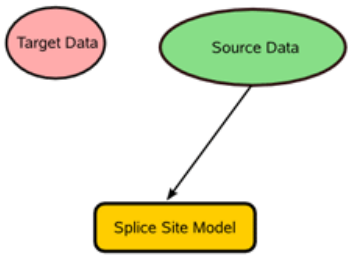
[Schweikert et al., 2008b; Widmer et al., 2010c]

Domain Adaptation Algorithms Overview

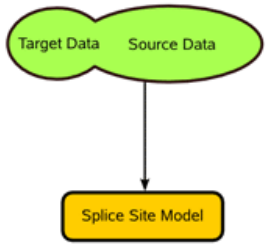
TargetOnly



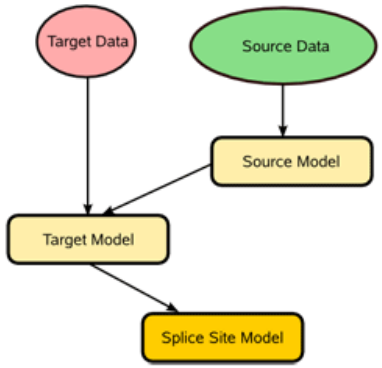
SourceOnly



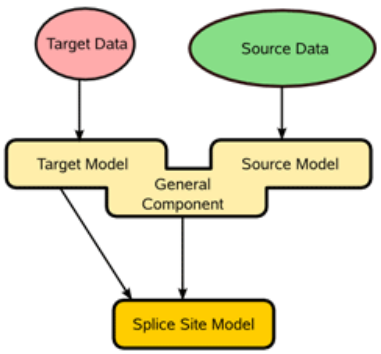
Combined



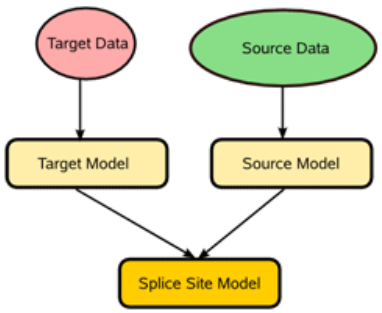
Regtrick



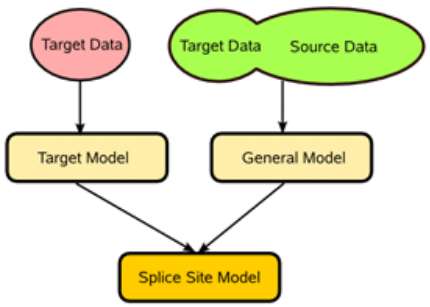
DualTask



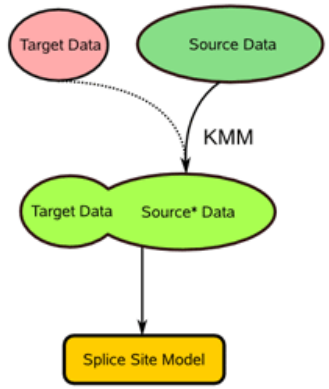
Convex



CombinedConvex

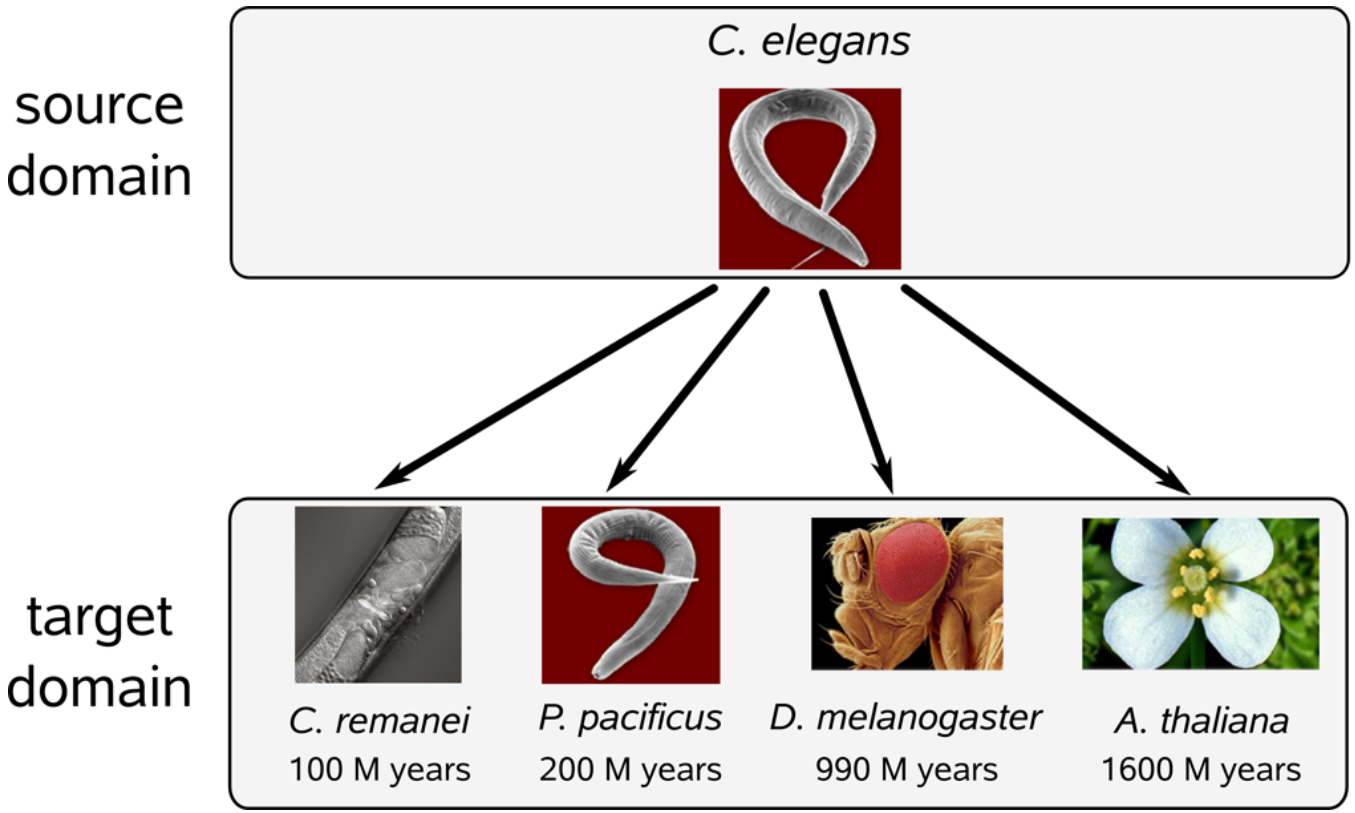


KMM/Shift



[Schweikert et al., 2008b]

Large-Scale Empirical Comparison



- Varying distances
- Different data set sizes

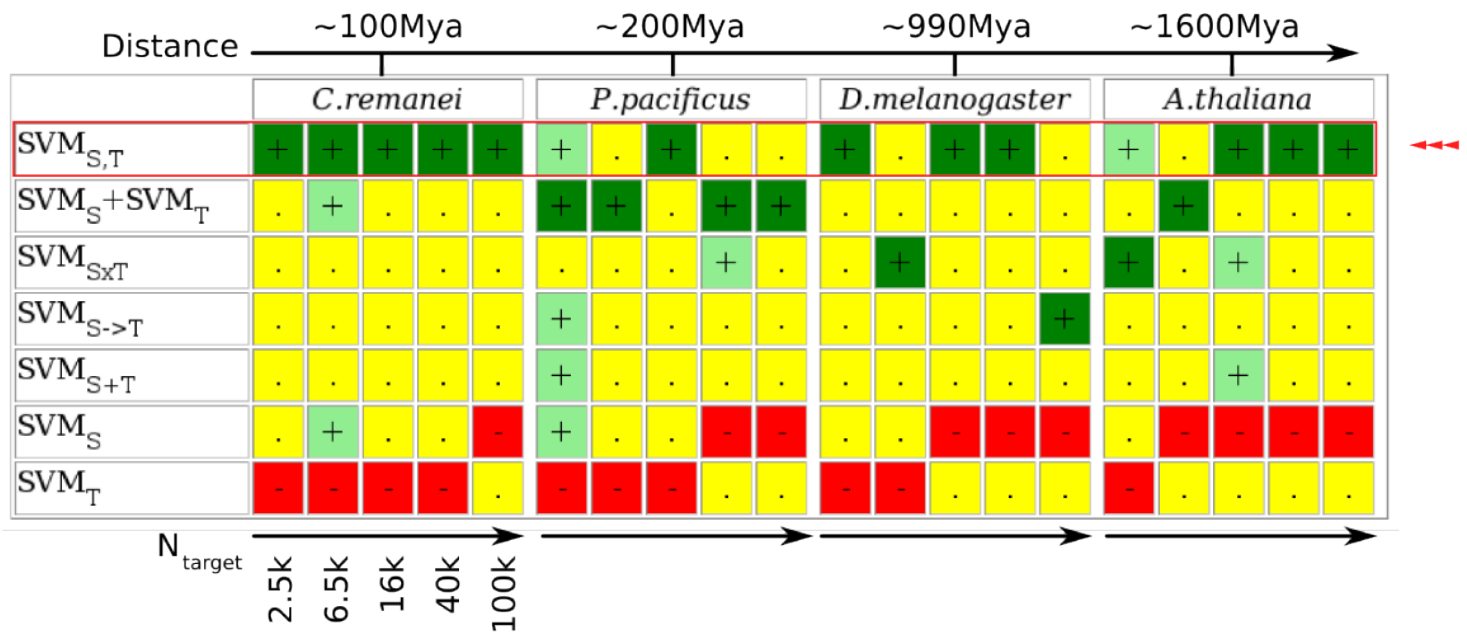
[MPI Developmental Biology and UCSC Genome Browser]

Experimental Setup

- Source dataset size: always 100k examples
- Target dataset sizes: $\{2500, 6500, 16000, 64000, 100000\}$
- Simple kernel (WDK of degree 1 $\Rightarrow$ under-fitting)
- Extensive model selection for each method
- Area under Precision/Recall curve for evaluation

Domain Adaptation Results Summary

- Considerable improvements possible
- Sophisticated domain adaptation methods needed on distantly related organisms
- Best overall performance has **DualTask**
- Most cost effective **Convex/AdvancedConvex**

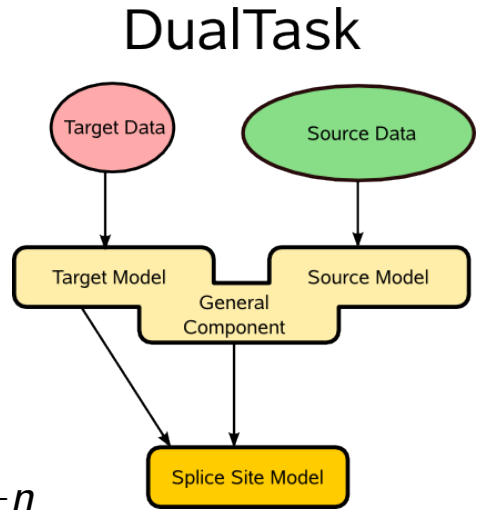


[Schweikert et al., 2008b]

Domain Adaptation Methods

Idea: [e.g., Caruana, 1997]

- Simultaneous optimization of both models
- Similarity between solution enforced



Approach:

$$\min_{w_S, w_T, \xi} \quad \frac{1}{2}\|w_S\|^2 + \frac{1}{2}\|w_T\|^2 - B w_T^T w_S + C \sum_{i=1}^{m+n} \xi_i$$

$$\text{s.t.} \quad y_i(\langle w_S, \Phi(x_i)\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, m$$

$$y_i(\langle w_T, \Phi(x_i)\rangle + b) \geq 1 - \xi_i \qquad i = m+1, \ldots, m+n$$

Equivalent to multi-task kernel learning: [Daume III, 2007]

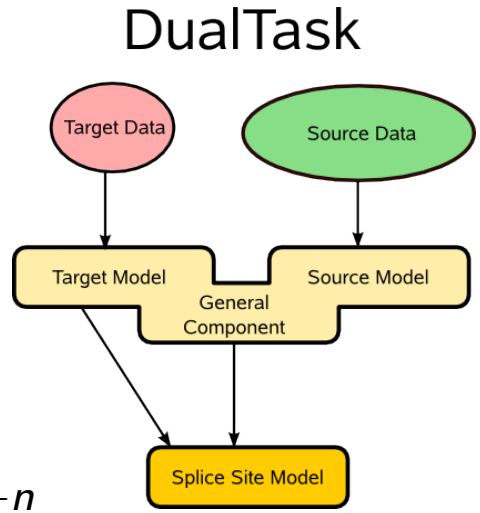
$$K_{MTK}((x,t),(x',t')) = \gamma_{t,t'} K(x,x')$$

for a suitably chosen Γ (p.s.d.).

Domain Adaptation Methods

Idea: [e.g., Caruana, 1997]

- Simultaneous optimization of both models
- Similarity between solution enforced



Approach:

$$\min_{w_S, w_T, \xi} \quad \frac{1}{2}\|w_S\|^2 + \frac{1}{2}\|w_T\|^2 - B w_T^T w_S + C \sum_{i=1}^{m+n} \xi_i$$

$$\text{s.t.} \quad y_i(\langle w_S, \Phi(x_i)\rangle + b) \geq 1 - \xi_i \qquad i = 1, \ldots, m$$

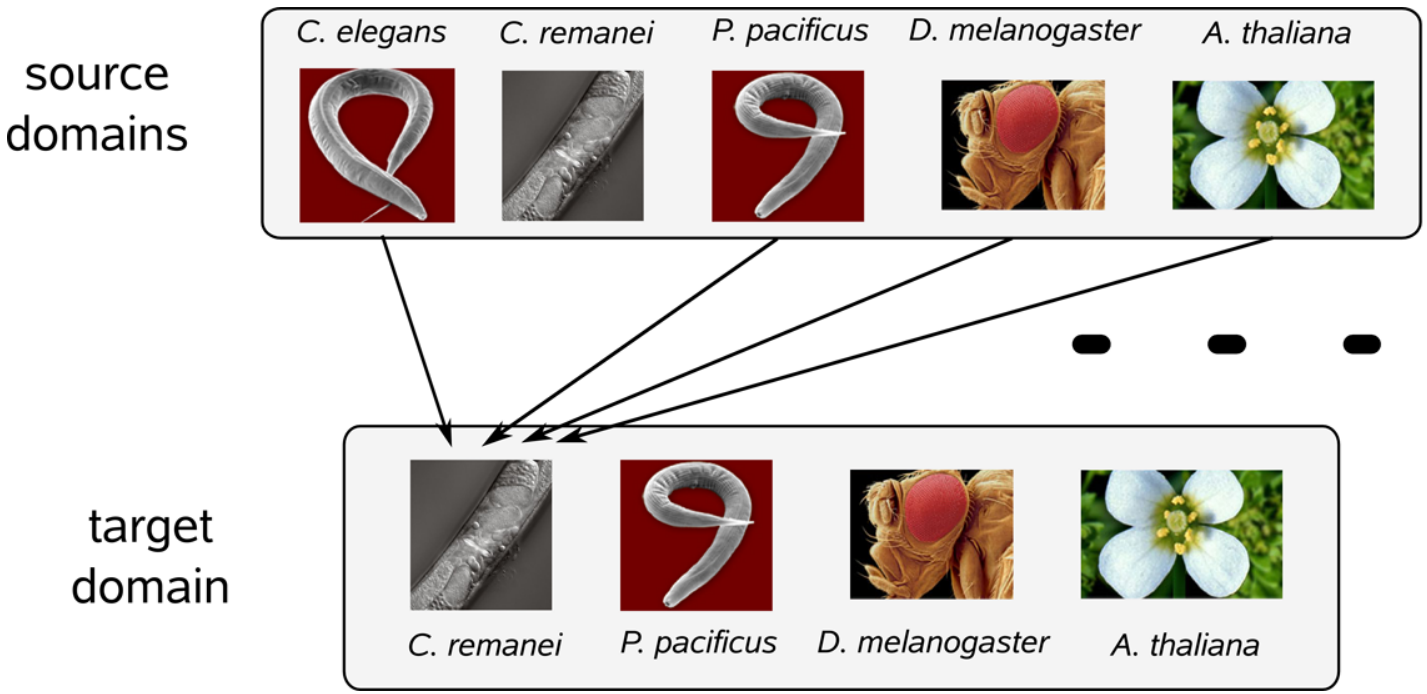
$$y_i(\langle w_T, \Phi(x_i)\rangle + b) \geq 1 - \xi_i \qquad i = m+1, \ldots, m+n$$

Equivalent to multi-task kernel learning: [Daume III, 2007]

$$K_{MTK}((x,t),(x',t')) = \gamma_{t,t'} K(x,x')$$

for a suitably chosen Γ (p.s.d.).

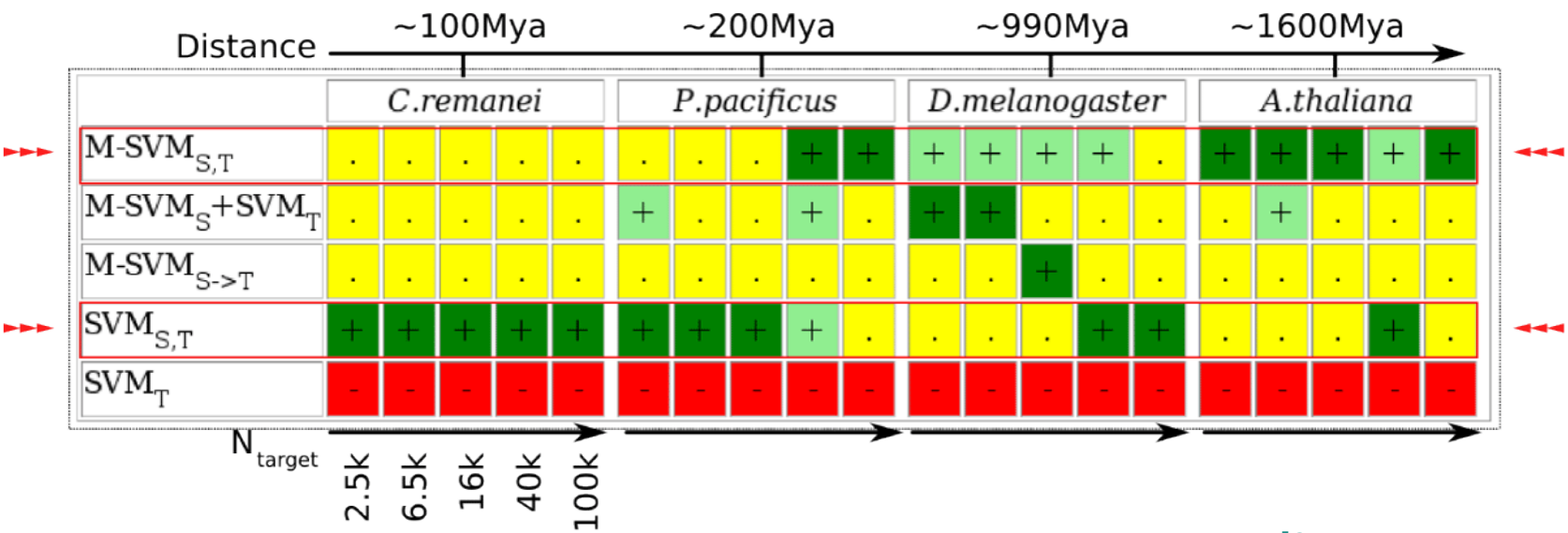
Multiple Source Domains



- Combine information from several sources (treated equally)
- Methods: Multi-task learning, Convex combination, Shifting

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest

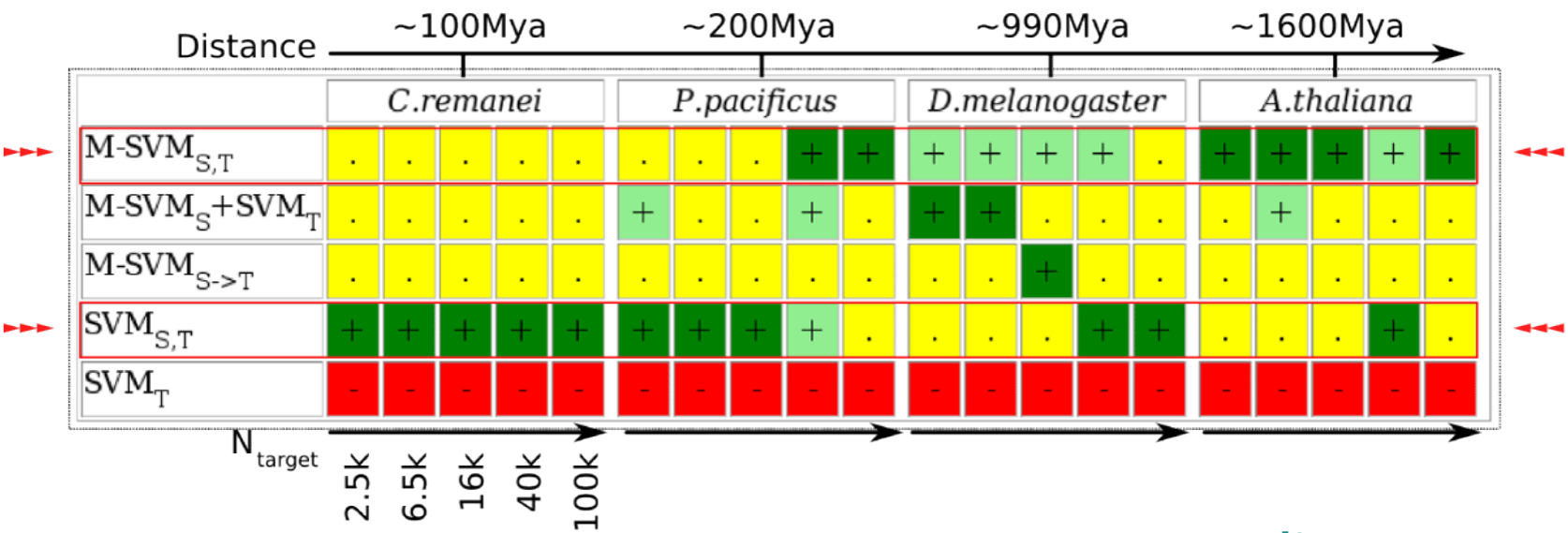


[Schweikert et al., 2008b]

Multiple sources can be worse than single source.
How to use information on relatedness in learning?

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest



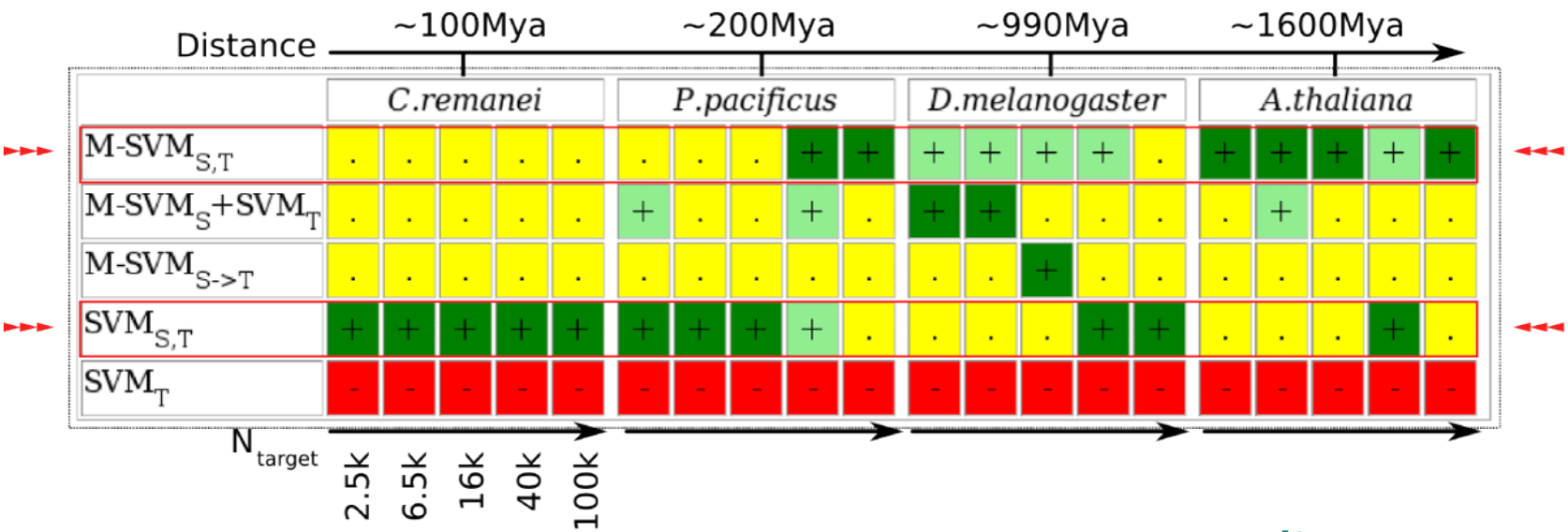
[Schweikert et al., 2008b]

Multiple sources can be worse than single source.

How to use information on relatedness in learning?

Results - Multiple Source Domains

- Single source model best for very closely related task
- Multiple source model better for distantly related tasks
- Multi-task algorithm strongest



[Schweikert et al., 2008b]

Multiple sources can be worse than single source.
How to use information on relatedness in learning?

Multitask learning

Hierarchical structure arises naturally from the Tree of Life



- Taxonomy defines relationship between tasks
- Closer tasks benefit more from each other

[Widmer et al., 2010b]

Two ways of leveraging a given taxonomy $\mathcal{T}$

Taxonomy



Transformation

Task Similarity Matrix

$$\Gamma = \begin{pmatrix} \gamma_{1,1} & \cdots & \gamma_{1,M} \\ \vdots & & \vdots \\ \gamma_{M,1} & \cdots & \gamma_{M,M} \end{pmatrix}$$

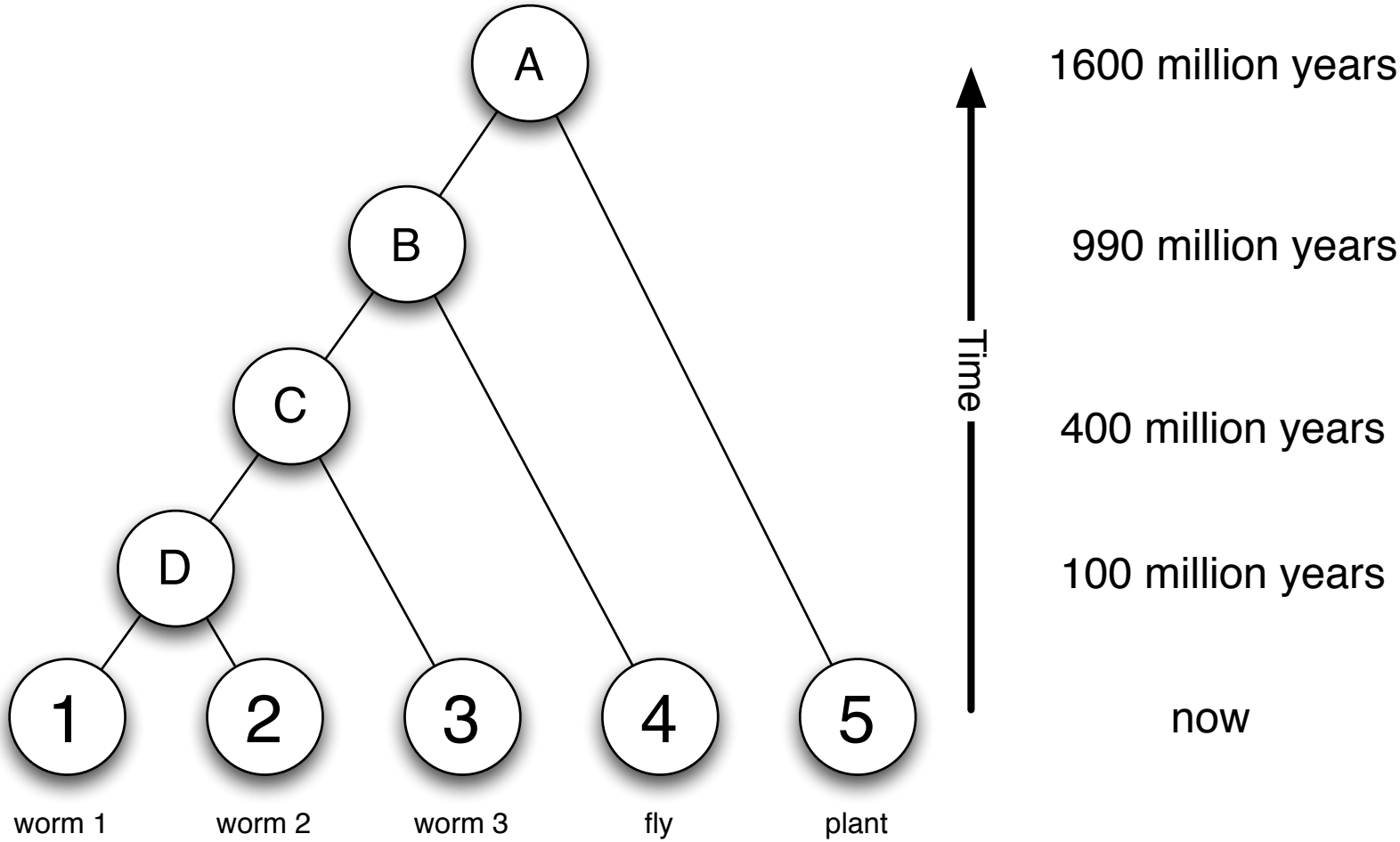
Top-Down

Multitask Kernel

$$K_{MTL}((x,t),(x',t')) = \gamma_{t,t'} K(x,x')$$

[Widmer et al., 2010b]

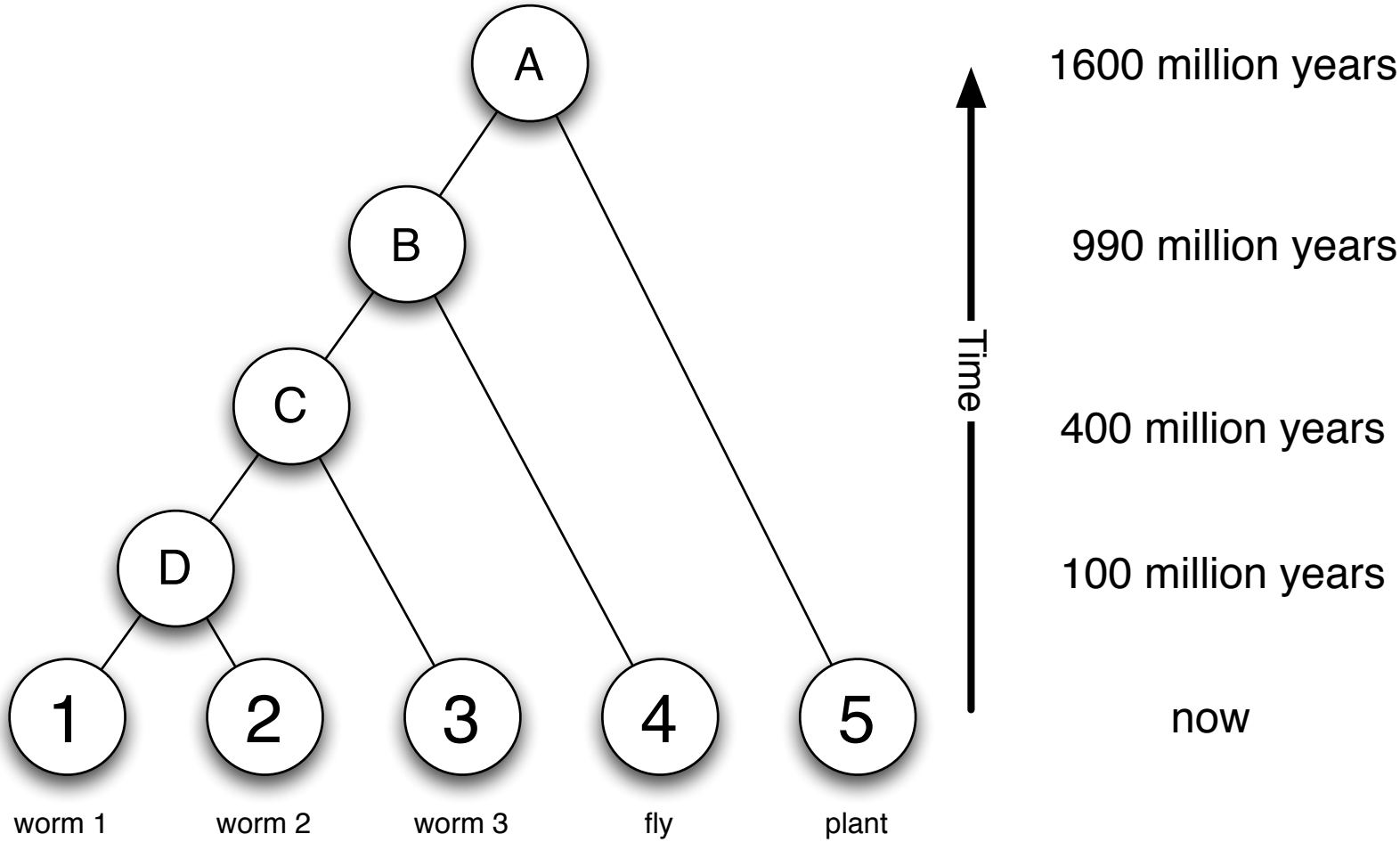
From Taxonomy to Γ



Idea: $\gamma_{i,j}$ should be inversely related to time to last common ancestor

Strategies: 1/years, **Hop-distance**, ...

From Taxonomy to Γ



Idea: $\gamma_{i,j}$ should be inversely related to time to last common ancestor

Strategies: 1/years, **Hop-distance**, ...

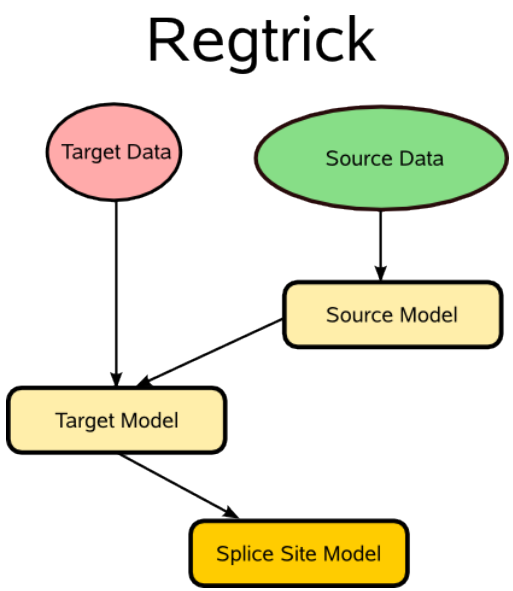
Hierarchical Top-Down Approach

Idea: Exploit taxonomy G in a top-down fashion

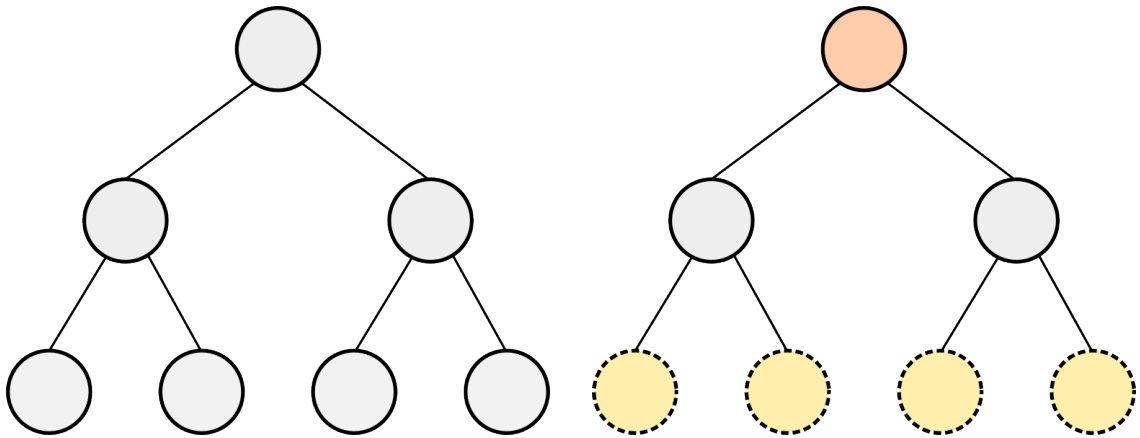
- Use taxonomy $\mathcal{T}$ in a top-down procedure
- Initialization: w_0 trained on union of all task datasets
- Top-Down for each node t:
  - Train on $D_i = \bigcup_{j \preccurlyeq i} D_j$
  - Regularize w_i against parent predictor w_{parent}:

$$\min_{w_i, b} \quad \frac{1}{2}\|w_i - w_{parent}\|^2 + C \sum_{(x,y)\in D_i} \ell\left(\langle \Phi(x), w_i \rangle + b, y\right),$$

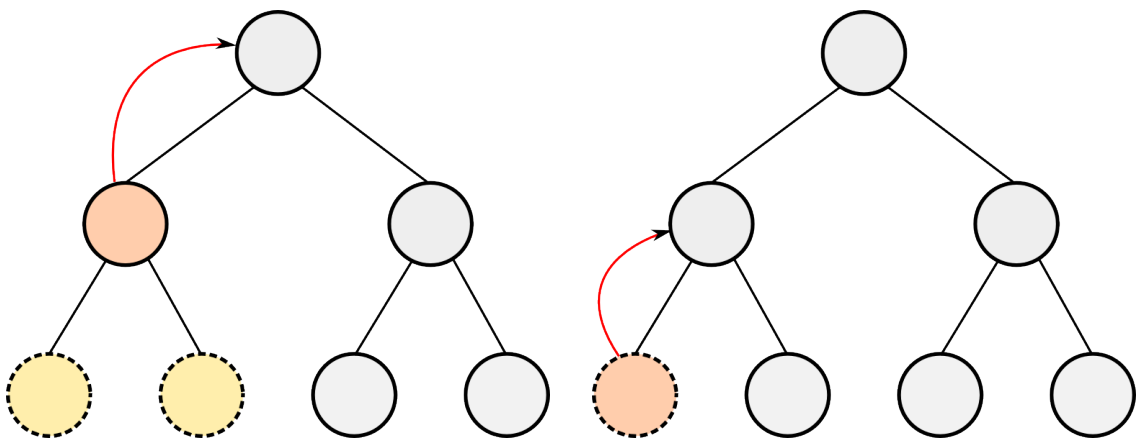
- Use leaf predictors for classification



Hierarchical Top-Down Approach: Illustration



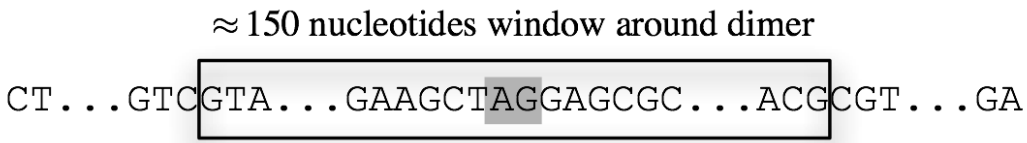
(a) Given taxonomy (b) Top-level training



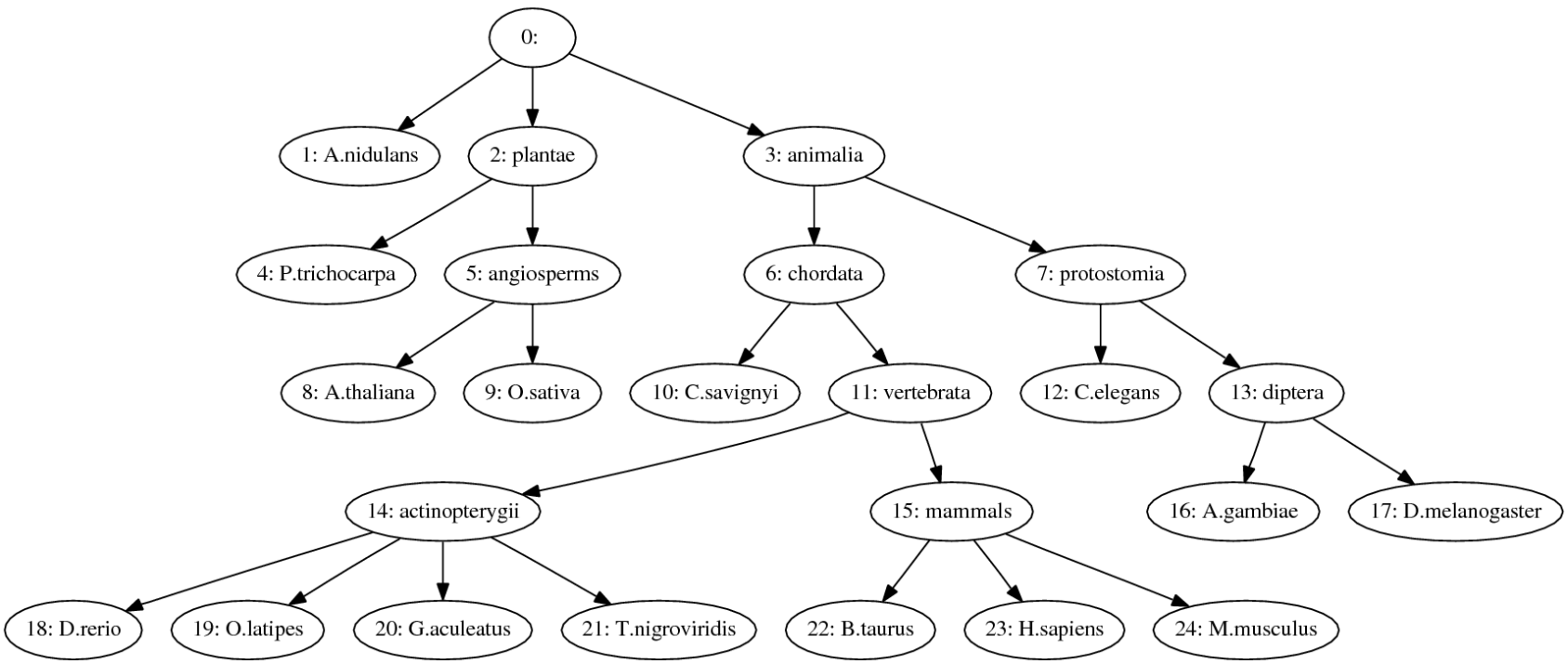
(c) Intermediate training (d) Taxon training

Application to Splicing Data

- Formulation as binary classification problem



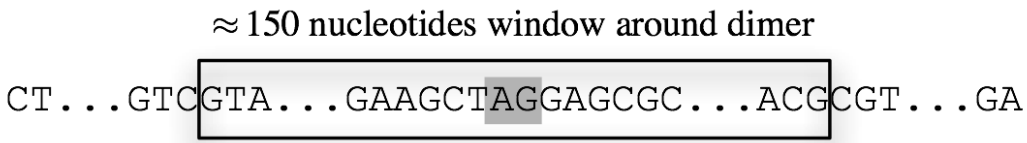
- Utilize 15 organisms related by taxonomy



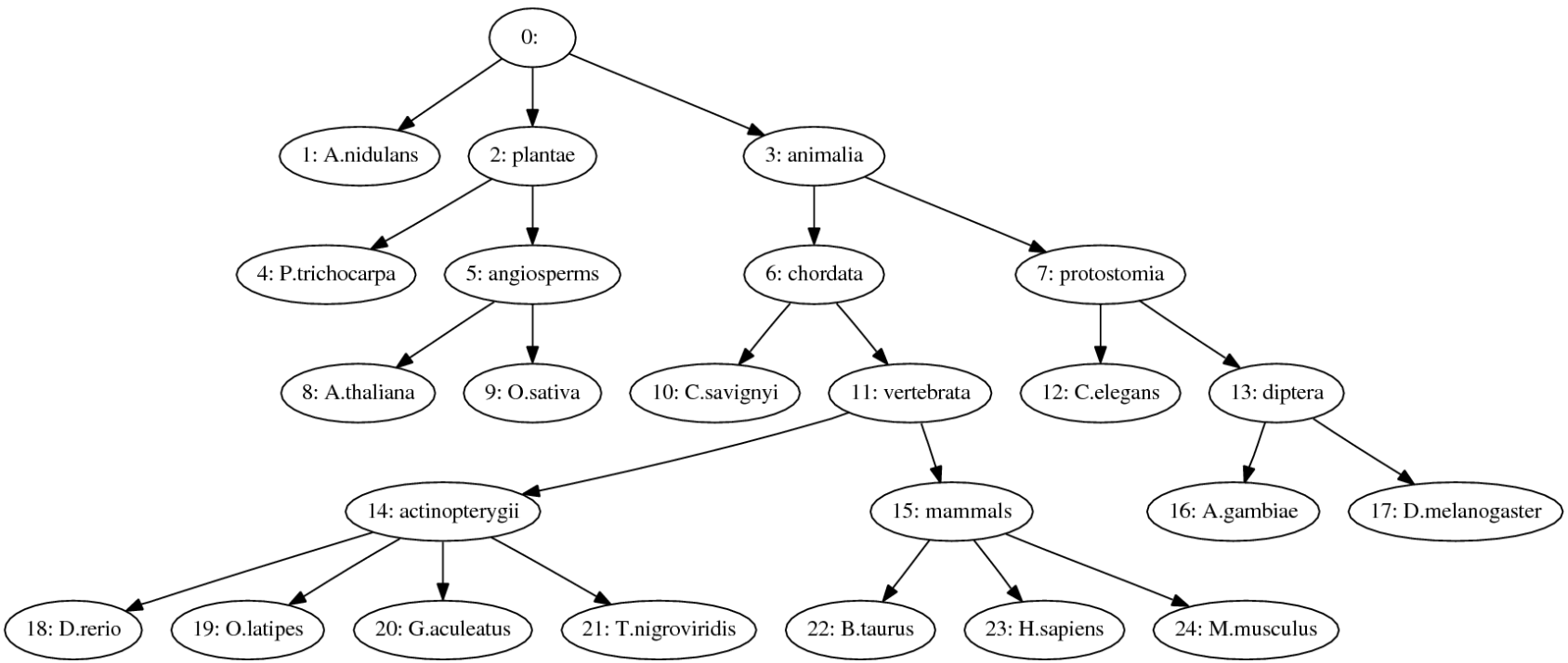
- Restricted to at most 10.000 examples per organism

Application to Splicing Data

- Formulation as binary classification problem

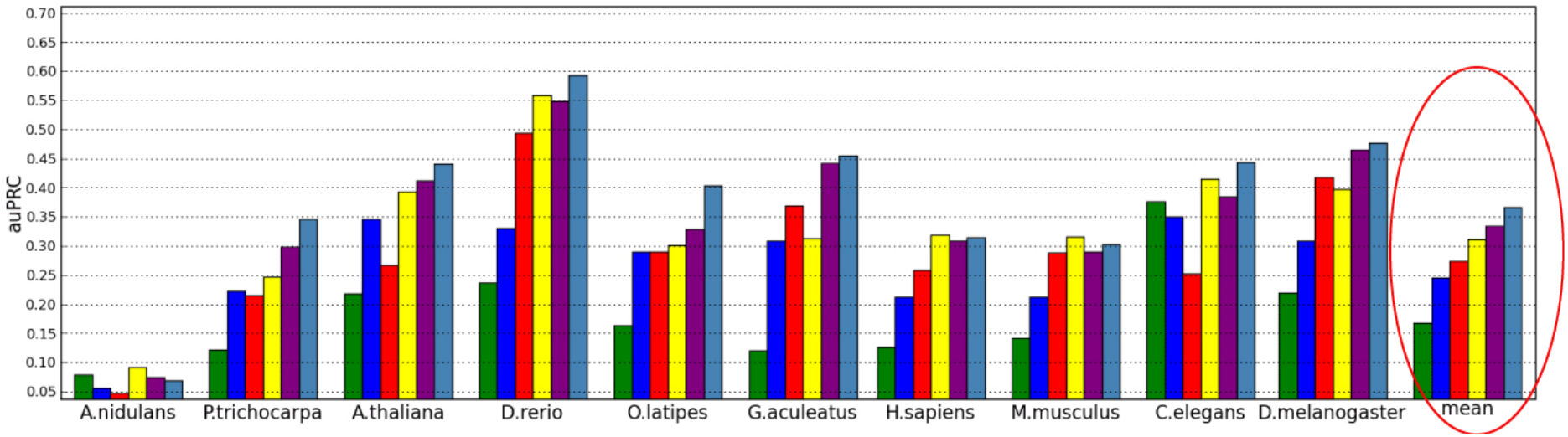


- Utilize 15 organisms related by taxonomy



- Restricted to at most 10.000 examples per organism

Results: Splicing Data



Observations:

- Union > Plain → conservation
- Often: Union > Nearest
- MTL methods outperform baselines
- Best performer: Top-Down (& MT-Kernel)

Discussion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

$$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

→ Multiple-Kernel Multi-Task Learning!

Discussion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

$$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

$\Rightarrow$ Multiple-Kernel Multi-Task Learning!

Discussion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

    $$\gamma_{i,j} = f(\text{“years of evolution between i and j”})?$$

Adaptive Multi-Task approach?

$\Rightarrow$ Multiple-Kernel Multi-Task Learning!

Discussion

- Hierarchy helps transferring information into right places
- Top-down approach transfers information most accurately
- Performance depends strongly on task similarity matrix
  - Choice very difficult, not easily done with cross-validation
  - Can we learn, e.g.

    $\gamma_{i,j} = f(\text{“years of evolution between i and j”})$?

    Adaptive Multi-Task approach?

    $\Rightarrow$ **Multiple-Kernel Multi-Task Learning!**

Multiple Kernel Learning with Meta-tasks



Figure : Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

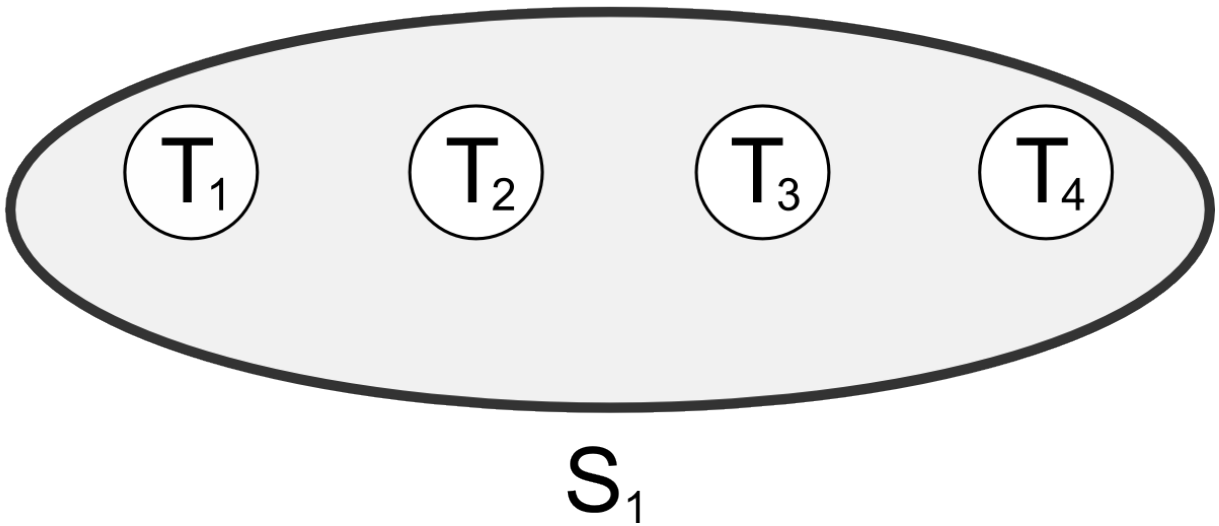


Figure : Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

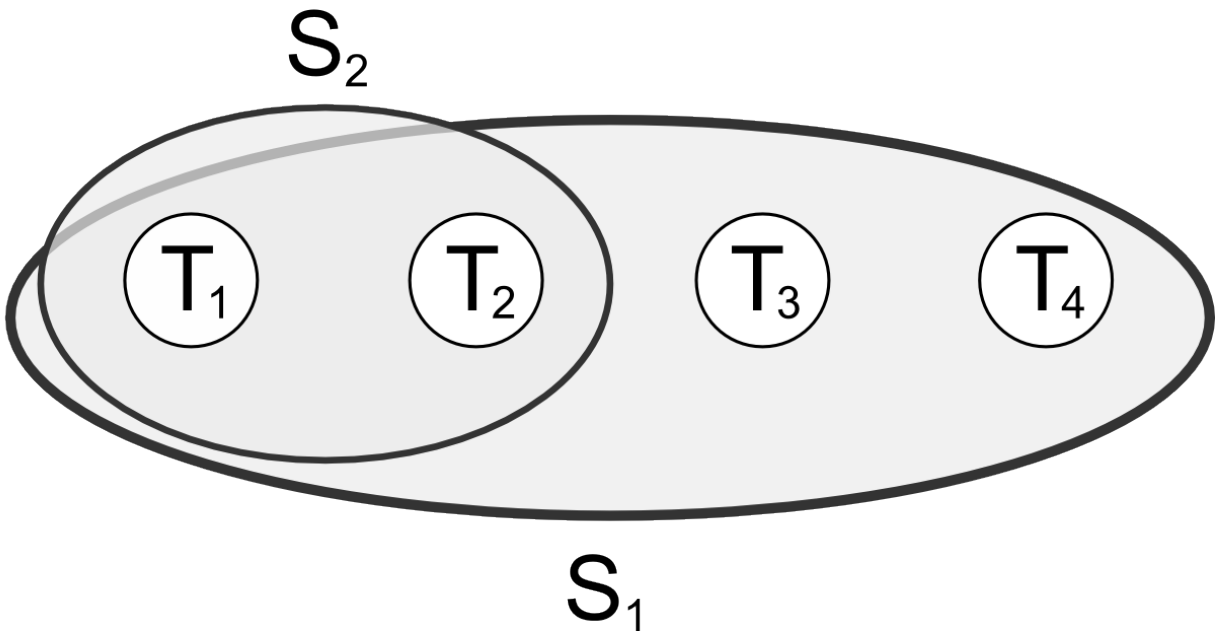


Figure : Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

Multiple Kernel Learning with Meta-tasks

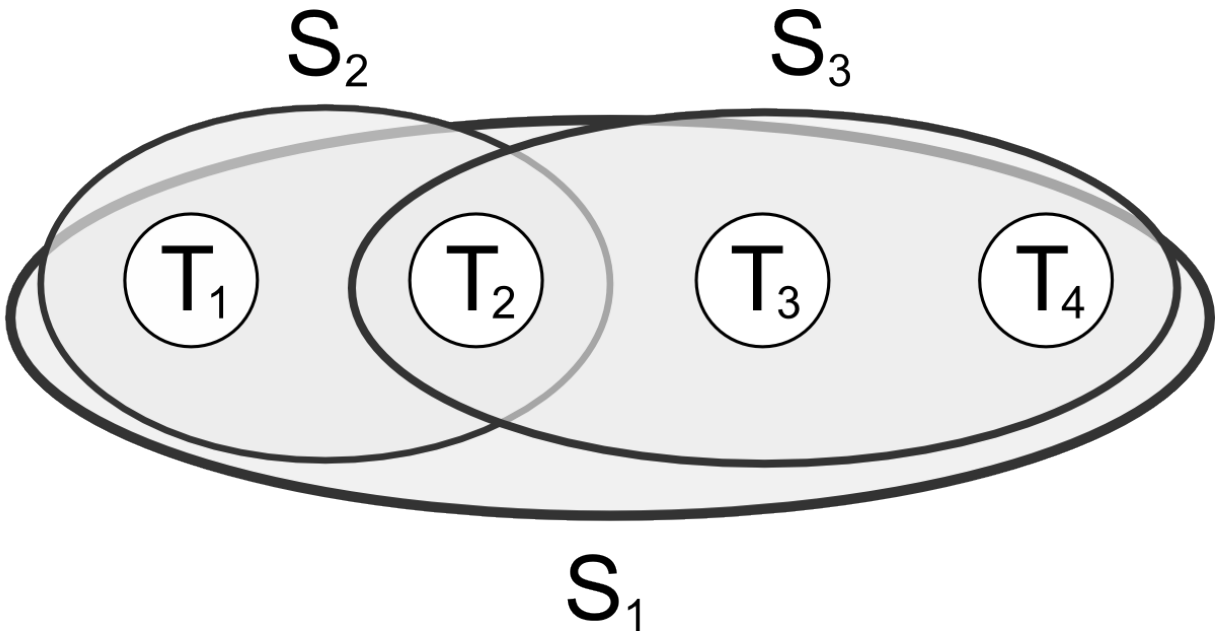


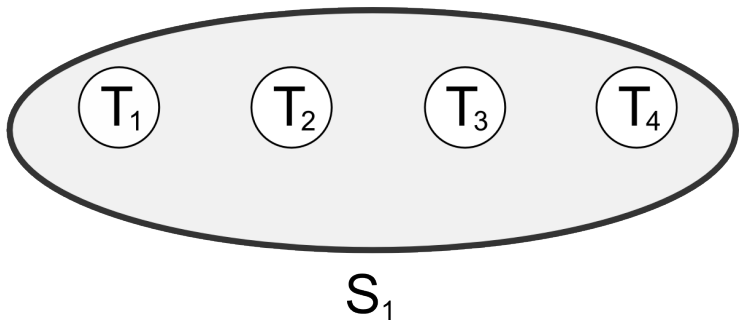
Figure : Generalization to meta-tasks.

- We use concept of meta-tasks to describe task-relationships
- Meta-task S captures shared property between sub-set of tasks
- The collection of meta-tasks $\mathcal{I}$ captures task structure

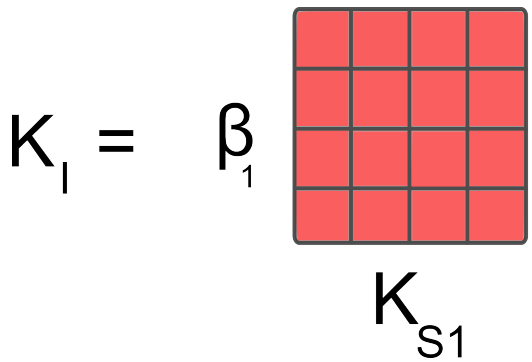
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



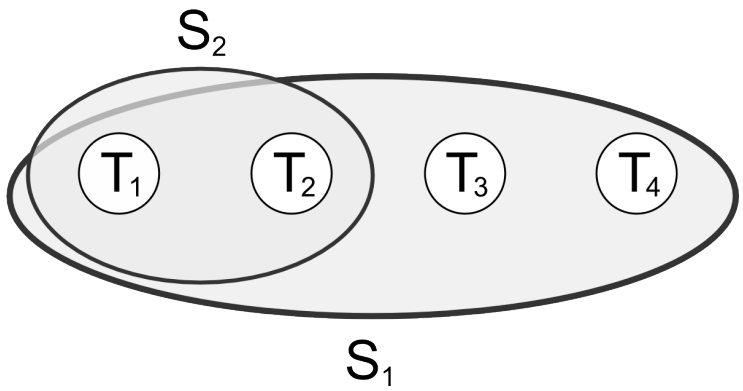
Example for collection of meta-tasks:



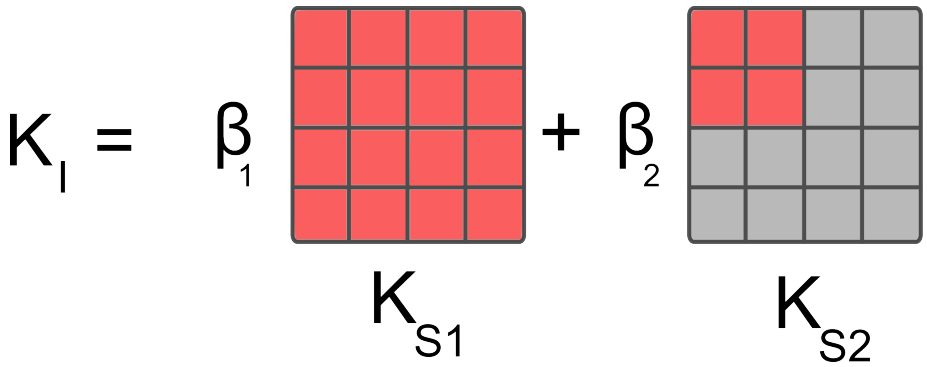
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



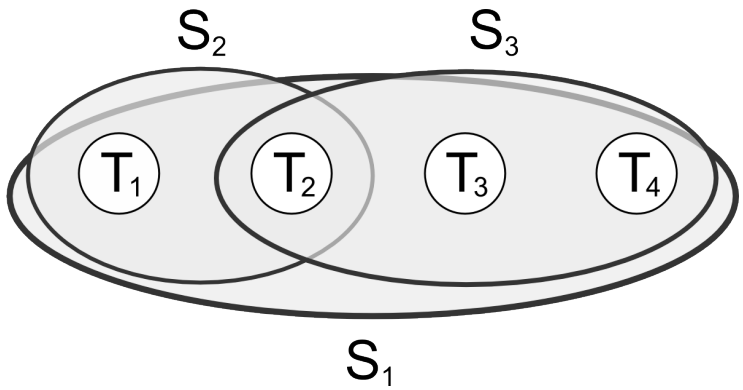
Example for collection of meta-tasks:



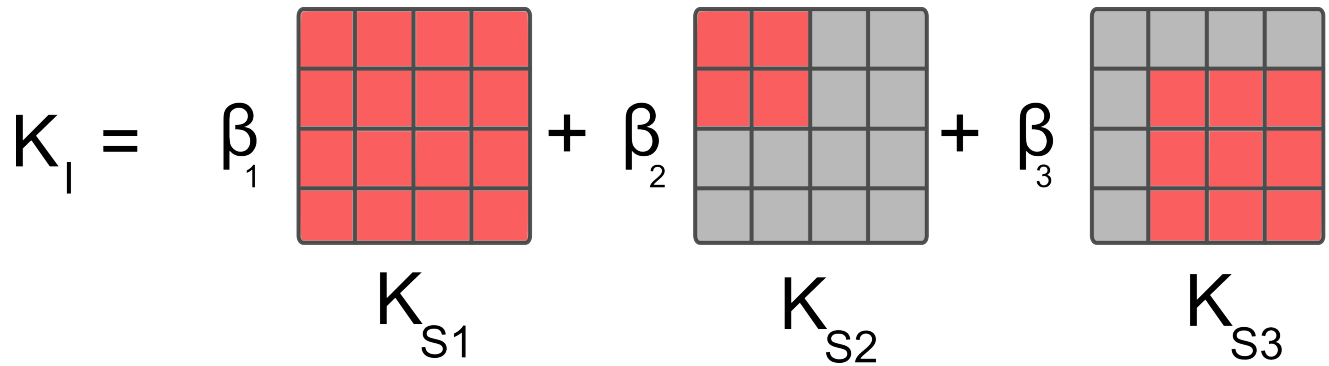
Decomposition of kernel matrix

$$K_S(x,y) = \begin{cases} K_B(x,y), & \text{if } \text{task}(x) \in S \wedge \text{task}(y) \in S \\ 0, & \text{else} \end{cases}$$

Thus, K_S defines kernel w.r.t. meta-task S



Example for collection of meta-tasks:



Optimization strategy: q-norm MKL

We use the MKL formulation: [Kloft et al., 2009]

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}(x_i, x_j)$$

$$\text{s.t.} \quad ||\boldsymbol{\beta}||_q^q \leq 1, \boldsymbol{\beta} \geq 0$$

$$\mathbf{Y}^T\boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Optimization strategy: q-norm MKL

We use the MKL formulation: [Kloft et al., 2009]

$$\min_{\boldsymbol{\beta}} \max_{\boldsymbol{\alpha}} \quad \mathbf{1}^T\boldsymbol{\alpha} - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}(x_i, x_j)$$

$$\text{s.t.} \qquad ||\boldsymbol{\beta}||_q^q \leq 1, \boldsymbol{\beta} \geq 0$$

$$\mathbf{Y}^T\boldsymbol{\alpha} = 0, 0 \leq \boldsymbol{\alpha} \leq C$$

- We learn the weights $\sum_{t=1}^{|\mathcal{I}|}\beta_t K_{S_t}$
- q lets us choose the appropriate norm (sparse/non-sparse)

$\Rightarrow$ How to define collection $\mathcal{I}$ of meta-tasks?

Power-set based approach

If no prior information available:

$$\mathcal{I}_{\mathcal{P}} = \{S | S \in \mathcal{P}(\mathcal{T}) \wedge S \neq \emptyset\}$$

- Consider Powerset $\mathcal{P}(\mathcal{T})$
- Most meta-tasks in power-set will be meaningless
  $\rightarrow$ learn sparse weights: $q = 1$
- Approach can be used to identify task structure **ab initio**
- 2^M meta-tasks $\rightarrow$ computationally expensive

Hierarchical decomposition

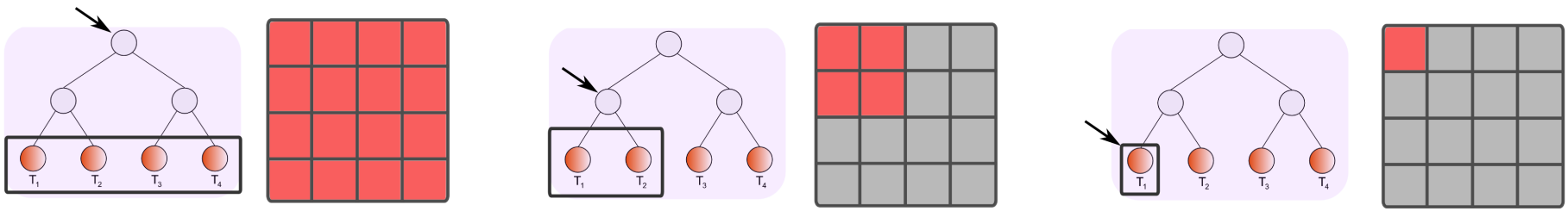
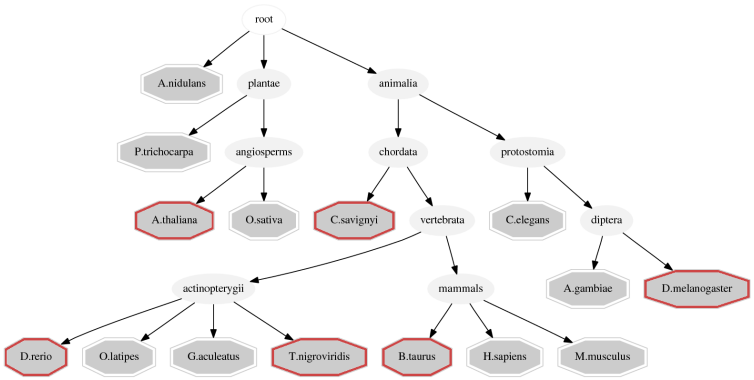


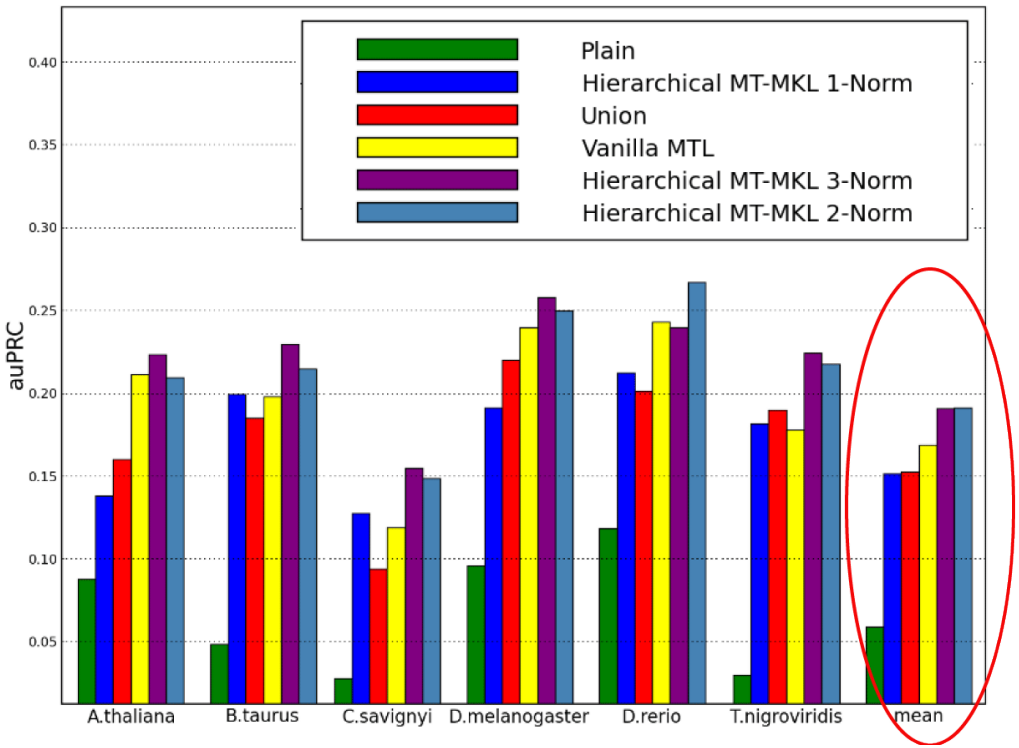
Figure : Example of taxonomy-based decomposition.

$$\mathcal{I}_{\mathcal{G}} = \{leaves(node) | node \in \mathcal{G}\}$$

- Meta-tasks are defined by taxonomy $\mathcal{G}$
- Taxonomy $\mathcal{G}$ gives us reasonable groups
  - Idea is to refine structure
  - Non-sparse combination ($q > 1$) for groupings

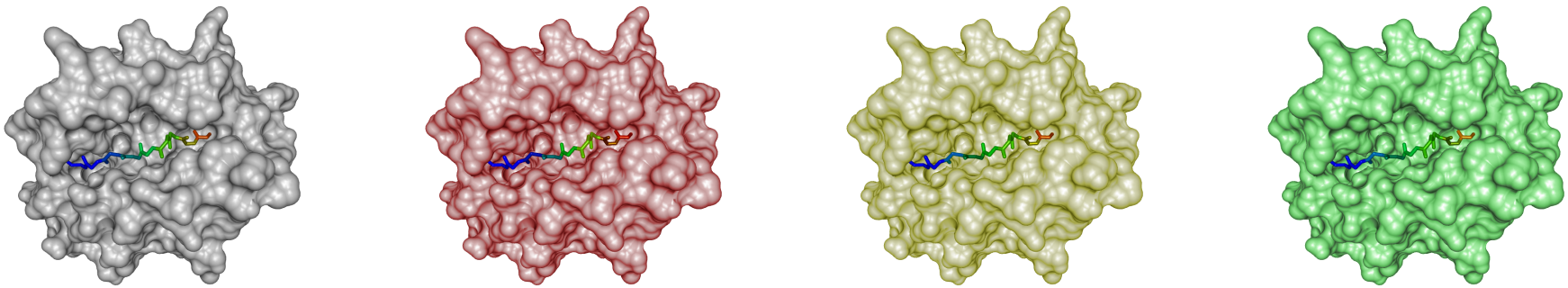
Experiments (a): Splice-site recognition





- Taxonomy is used to define collection of meta-tasks $\mathcal{I}$
- Baselines: Plain, Union, Vanilla MTL
- Best performance for norm $q = 2, 3$

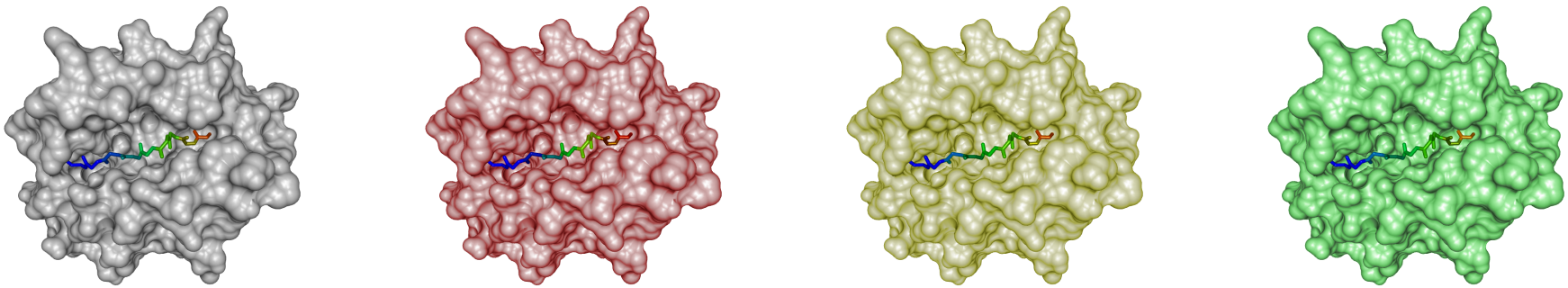
Experiments (b): MHC-I binding prediction



Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	**69.9%**

- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

Experiments (b): MHC-I binding prediction

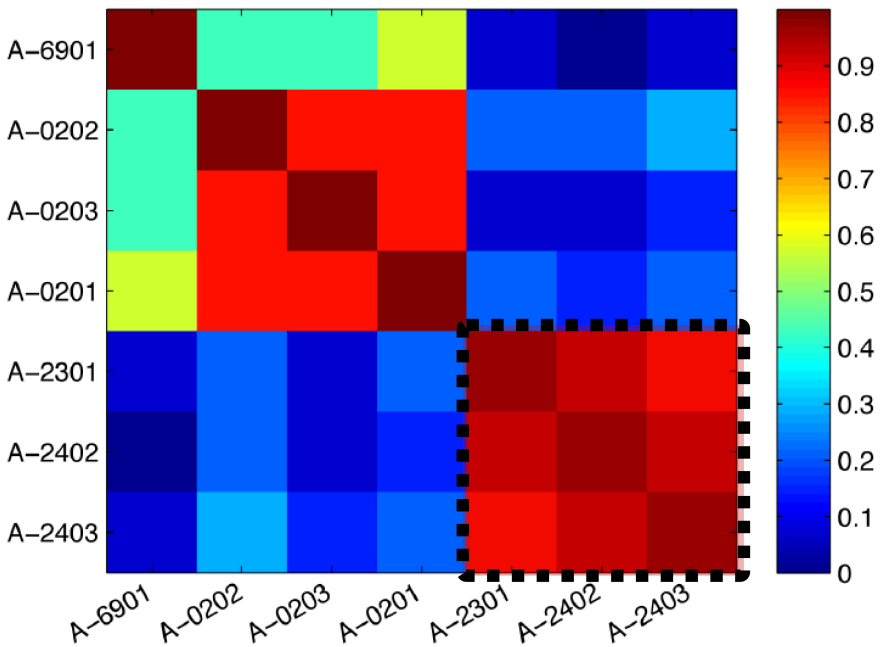


Method	Plain	Union	Vanilla MTL	Powerset MT-MKL
auPRC	67.1%	57.6%	67.9%	**69.9%**

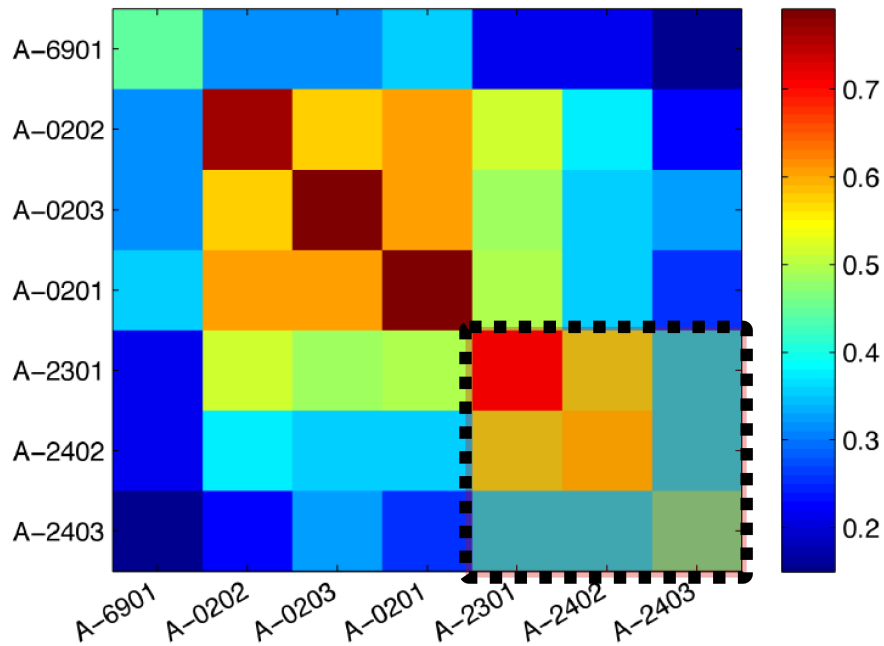
- No task structure (used): Powerset MT-MKL
- Question: Can we identify meaningful structure?

Experiments (b): MHC-I binding prediction

Learned weights can also be used for interpretation purposes:



(a) Sequence similarity between alleles



(b) Task similarity computed from weights

- Similarity computed from meta-task weights
- Comparison to similarity between peptide sequences
- Successfully identifies biological meaningful structure

Solving Large-Scale MT-SVMs

So far we solved this optimization problem:

$$\max_{\alpha} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K((x_i,s_i),(z,t_i)) + \sum_{i=1}^{n}\alpha_i$$

$$\text{s.t. } 0 \le \alpha_i \le C \;\forall i \in [1,n]$$

$$\alpha^T \mathbf{y} = 0,$$

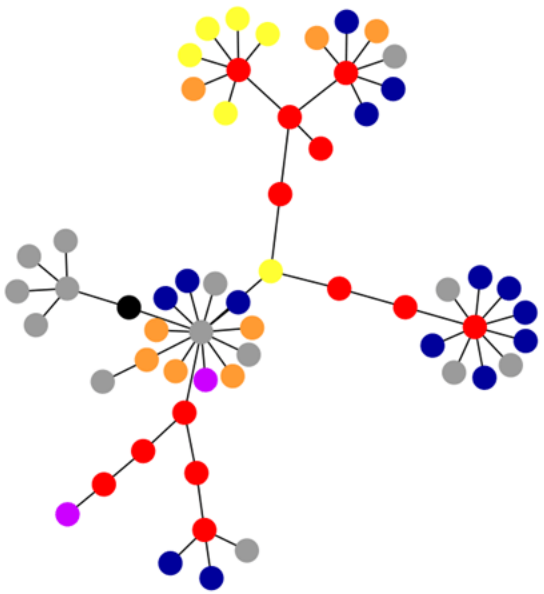
using the following choice of Multitask Kernel

$$K((x,s),(z,t)) = \gamma_{s,t} \cdot K_{\text{examples}}(x,z).$$

- Readily plugged into existing solvers (e.g. LibSVM, SVMLight)
- Suffers from problems of dual SVM solvers (e.g. $n > d$)

$\Rightarrow$ Fails to exploit recent advances in Linear SVM solvers!

Graph-based MTL



- Not all tasks are equally similar
- $\Rightarrow$ need weighting of tasks
- achieved by *graph-based MTL*

Graph-based MTL (Evgeniou et al. [2005]): given a graph adjacency matrix $A = (A_{s,t})$, promote similar weights w_s, w_t for similar tasks s, t, i.e., minimize

$$J(w_1, ..., w_T) = \sum_{s=1}^{T} \sum_{t=1}^{T} \|w_s - w_t\|^2 A_{s,t} \, .$$

Equivalent Formulations for $L = \Gamma^+$

Can be written in terms of the graph Laplacian L corresponding to the graph adjacency matrix A: [Evgeniou et al., 2005]

$$J(w_1, \ldots, w_T)_{\text{graph}} = \sum_s \sum_t \|w_s - w_t\|^2 A_{s,t} = \sum_s \sum_t w_s^T w_t L_{s,t}$$

(Laplacian here defined as $L = D - A$ with $D_{s,t} := \delta_{s,t} \sum_u A_{s,u}$.)

Graph-based multi-task learning:

Given a convex loss function l,

$$\min_{w_1,\ldots,w_T \in \mathbb{R}^m} \underbrace{\frac{1}{2} \sum_{t=1}^{T} \|w_t\|_2^2}_{\text{standard regularizer}} + \underbrace{\frac{1}{2} \sum_{s=1}^{T} \sum_{t=1}^{T} L_{st} w_s^\top w_t}_{\textbf{multi-task regularizer}} + \underbrace{C \sum_{i=1}^{n} l\left(y_i w_{t(i)}^\top x_i\right)}_{\text{empirical loss}}$$

Novel Block View

We define $w = (w_1^\top, \ldots, w_T^\top)^\top$ and $\psi(x_i) = (0, \ldots, x_i, \ldots, 0)^\top$

$$\text{block}(B) := \begin{pmatrix} \text{diag}(b_{11}) & \cdots & \text{diag}(b_{1T}) \\ \vdots & & \vdots \\ \text{diag}(b_{T1}) & \cdots & \text{diag}(b_{TT}) \end{pmatrix} .$$

Generalized primal MTL problem *(block view)*

$$\min_w \quad \frac{1}{2} w^\top \text{block}(I + L) w + C \sum_{i=1}^{n} l\left(y_i w^\top \psi(x_i)\right) ,$$

where I is the identity matrix in $\mathbb{R}^{T \times T}$.

$\Rightarrow$ Use Fenchel Duality to compute general dual!

[Widmer et al., 2012]

Novel Block View

We define $w = (w_1^\top, \ldots, w_T^\top)^\top$ and $\psi(x_i) = (0, \ldots, x_i, \ldots, 0)^\top$

$$\text{block}(B) := \begin{pmatrix} \text{diag}(b_{11}) & \cdots & \text{diag}(b_{1T}) \\ \vdots & & \vdots \\ \text{diag}(b_{T1}) & \cdots & \text{diag}(b_{TT}) \end{pmatrix} .$$

Generalized primal MTL problem *(block view)*

$$\min_w \quad \frac{1}{2} w^\top \text{block}(I + L) w + C \sum_{i=1}^{n} l\left(y_i w^\top \psi(x_i)\right) ,$$

where I is the identity matrix in $\mathbb{R}^{T \times T}$.

$\Rightarrow$ **Use Fenchel Duality to compute general dual!**

[Widmer et al., 2012]

Fenchel Duality

Extending techniques presented by Rifkin and Lippert [2007], we derive the Fenchel dual:

$$\max_{\alpha} \quad -C\sum_i l^*\Big(-\frac{\alpha_i}{C}\Big) - \frac{1}{2}\Big\|\sum_i \alpha_i y_i \psi(x_i)\Big\|^2_{\mathbf{block}(M)}$$

where

$$M := (I + L)^{-1}$$

	function	conjugate function
hinge loss	$\max(0, 1-t)$	t if $-1 \le t \le 0$ and ∞ else
ℓ_p-norm	$\frac{1}{2}\|w\|_p^2$	$\frac{1}{2}\|w\|_{p^*}^2$ where $p^* = \frac{p}{p-1}$
quadratic form	$\frac{1}{2} w^\top B w$	$\frac{1}{2} w^\top B^{-1} w$

$\Rightarrow$ SVM: Use conjugate of hinge loss!

Fenchel Duality

Extending techniques presented by Rifkin and Lippert [2007], we derive the Fenchel dual:

$$\max_{\alpha} \quad -C\sum_i l^*\Big(-\frac{\alpha_i}{C}\Big) - \frac{1}{2}\Big\|\sum_i \alpha_i y_i \psi(x_i)\Big\|^2_{\mathbf{block}(M)}$$

where

$$M := (I + L)^{-1}$$

	function	conjugate function
hinge loss	$\max(0, 1-t)$	t if $-1 \le t \le 0$ and ∞ else
ℓ_p-norm	$\frac{1}{2}\|w\|_p^2$	$\frac{1}{2}\|w\|_{p^*}^2$ where $p^* = \frac{p}{p-1}$
quadratic form	$\frac{1}{2} w^\top B w$	$\frac{1}{2} w^\top B^{-1} w$

$\Rightarrow$ SVM: Use conjugate of hinge loss!

Special Case: Large-Margin Learning

Denote by $M := (I + L)^{-1}$. Then the dual MTL-SVM problem is given by:

$$\max_{\mathbf{0} \leq \boldsymbol{\alpha} \leq \mathbf{C}} \quad \mathbf{1}^\top \boldsymbol{\alpha} - \frac{1}{2} \left\| \sum_i \alpha_i y_i \psi(x_i) \right\|^2_{\text{block}(M)}$$

- General formulation instantiated for hinge-loss

Dual Coordinate Descend

Idea: Optimize with respect to a single training example at a time

[Hsieh et al., 2008]

Denoting the task associated to example i by $t(i)$, the dual objective reduces to:

$$\underset{d:0\leq\alpha_i+d\leq C}{\operatorname{argmax}} \; d - \frac{1}{2}d^2 x_i^\top x_i - d w_{t(i)}^\top y_i x_i \tag{1}$$

where w is given by the KKT conditions:

Lemma (Representer theorem)

In the primal-dual optimal point, $w = \mathrm{block}(M)\sum_i \alpha_i y_i \psi(x_i)$.

Setting the gradient of Eq. (1) to zero gives rise to the following update rule

$$d = \frac{1 - w_{t(i)}^\top y_i x_i}{x_i^\top x_i}\,.$$

Training Algorithm

1: **input:** $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
2: for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
3: for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
4: **while** optimality conditions are not satisfied **do**
5: **for** all $i \in \{1, \ldots, n\}$
6: compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
7: store $\hat{\alpha}_i := \alpha_i$
8: put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
9: for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
10: **end for**
11: **end while**
12: **output:** $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

1: **input:** $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
2: for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
3: for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
4: **while** optimality conditions are not satisfied **do**
5: **for** all $i \in \{1, \ldots, n\}$
6: compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
7: store $\hat{\alpha}_i := \alpha_i$
8: put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
9: for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
10: **end for**
11: **end while**
12: **output:** $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

1: **input:** $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
2: for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
3: for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
4: **while** optimality conditions are not satisfied **do**
5: **for** all $i \in \{1, \ldots, n\}$
6: compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
7: store $\hat{\alpha}_i := \alpha_i$
8: put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
9: for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
10: **end for**
11: **end while**
12: **output:** $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

1: **input:** $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
2: for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
3: for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
4: **while** optimality conditions are not satisfied **do**
5: **for** all $i \in \{1, \ldots, n\}$
6: compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
7: store $\hat{\alpha}_i := \alpha_i$
8: put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
9: for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
10: **end for**
11: **end while**
12: **output:** $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

1: **input:** $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
2: for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
3: for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
4: **while** optimality conditions are not satisfied **do**
5: **for** all $i \in \{1, \ldots, n\}$
6: compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
7: store $\hat{\alpha}_i := \alpha_i$
8: put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
9: for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
10: **end for**
11: **end while**
12: **output:** $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

input: $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
while optimality conditions are not satisfied **do**
    for all $i \in \{1, \ldots, n\}$
        compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
        store $\hat{\alpha}_i := \alpha_i$
        put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
        for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
    end for
end while
output: $w_1, \ldots, w_T$

- We prove convergence of the above algorithm.

Training Algorithm

input: $x_1, \ldots, x_n \in \mathbb{R}^m$, $t(1), \ldots, t(n) \in \{1, \ldots, T\}$, $y_1, \ldots, y_n \in \{-1, 1\}$
for all $i \in \{1, \ldots, n\}$ initialize $\alpha_i = 0$
for all $t \in \{1, \ldots, T\}$ initialize $w_t = \mathbf{0}$
while optimality conditions are not satisfied **do**
    for all $i \in \{1, \ldots, n\}$
        compute step size d by update rule $d = 1 - w_{t(i)}^\top y_i x_i / x_i^\top x_i$
        store $\hat{\alpha}_i := \alpha_i$
        put $\alpha_i := \max(0, \min(C, \hat{\alpha}_i + d))$
        for all $s = 1, \ldots, T$, update $w_s := w_s + m_{s,t(i)}(\alpha_i - \hat{\alpha}_i) y_i x_i$
    end for
end while
output: $w_1, \ldots, w_T$

- **We prove convergence of the above algorithm.**

Computational Experiments

Data sets:

	#dim	#examples	#tasks
Gauss2D	2	$1 \cdot 10^5$	2
Breast Cancer	44	474	3
MNIST-MTL	784	$9.0 \cdot 10^3$	3
Land Mine	9	$1.5 \cdot 10^4$	29
Splicing	$6 \cdot 10^6$	$6.4 \cdot 10^6$	4

- Different dimensionality
- Different data set sizes
- Different number of tasks

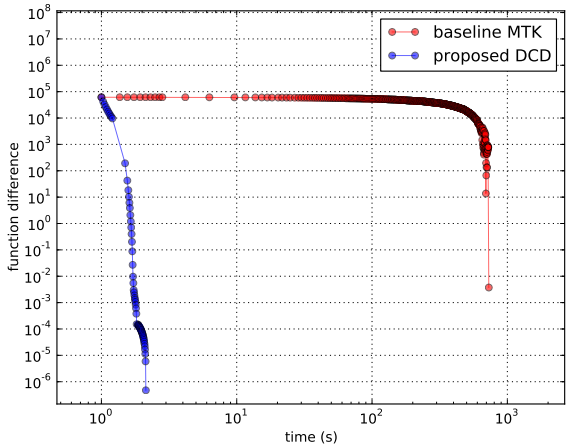
Computational Experiments

Data sets:

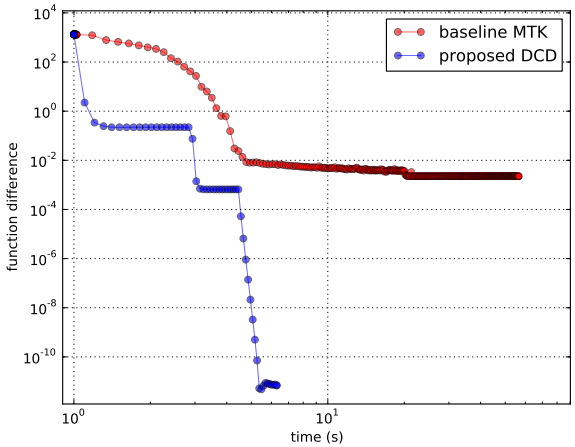
	#dim	#examples	#tasks
Gauss2D	2	$1 \cdot 10^5$	2
Breast Cancer	44	474	3
MNIST-MTL	784	$9.0 \cdot 10^3$	3
Land Mine	9	$1.5 \cdot 10^4$	29
Splicing	$\mathbf{6 \cdot 10^6}$	$\mathbf{6.4 \cdot 10^6}$	**4**

- Different dimensionality
- Different data set sizes
- Different number of tasks

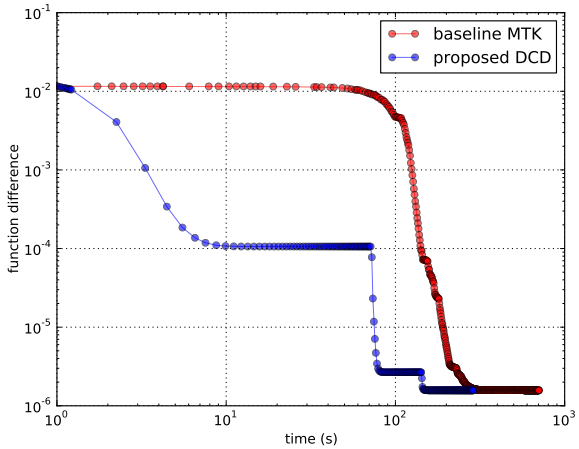
Results: Convergence



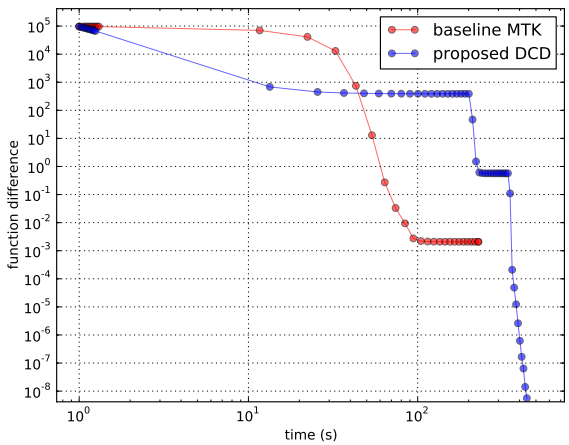
(a) Gauss2D



(b) Breast cancer

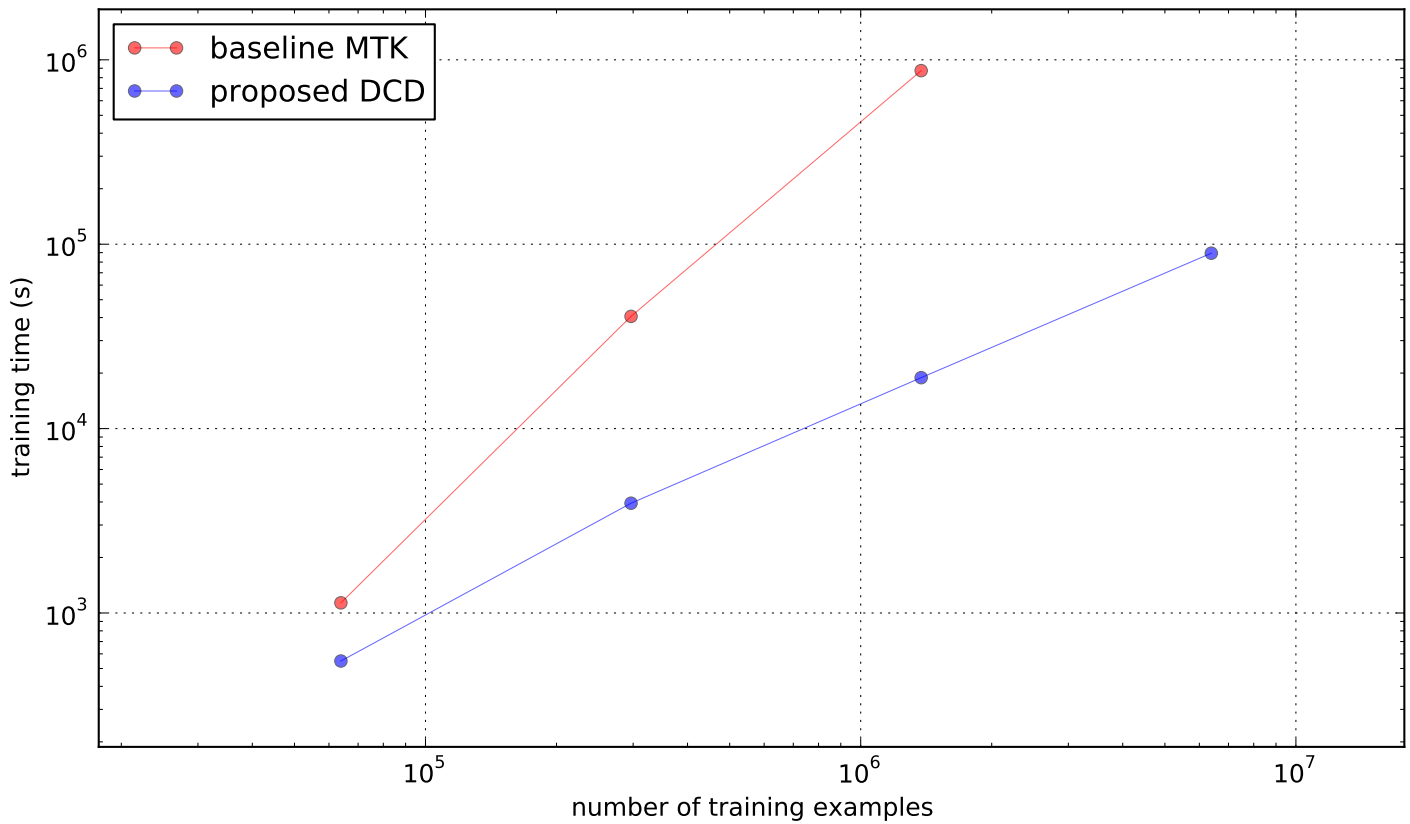


(c) MNIST-MTL



(d) Land Mine

Results: Large-scale Experiment



- COFFIN by Sonnenburg and Franc [2010] for encoding high-dimensional sparse feature vectors as dense lower dimensional vectors

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: `www.raetschlab.org/suppl/transfer-learning`

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: `www.raetschlab.org/suppl/transfer-learning`

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: `www.raetschlab.org/suppl/transfer-learning`

Summary

Domain adaptation:

- Considerable improvements possible
- Sophisticated methods slight edge for distantly related tasks

Multitask learning:

- Novel methods provide scalable way of integrating information
  (Implementations available for SVMLight and LibSVM)
- Design of task similarity matrix critical & difficult

Recent extensions:

- Estimation of "optimal" task similarity matrix
- Extension to structured output learning
- Cleaner formulations, Large-scale MTK-MT-SVMs

Material available at: `www.raetschlab.org/suppl/transfer-learning`

Acknowledgements



Christian Widmer

(MSKCC & TU Berlin)



Marius Kloft

(MSKCC & NYU)

Involved earlier

- Gabriele Schweikert
- Nico Görnitz
- Nora Toussaint
- Jose Leiva
- Yasemin Altun
- Bernhard Schölkopf

Funding by German Research Foundation, Max Planck Society & MSKCC

Thank you for your attention!

Acknowledgements



Christian Widmer

(MSKCC & TU Berlin)



Marius Kloft

(MSKCC & NYU)

Involved earlier

- Gabriele Schweikert
- Nico Görnitz
- Nora Toussaint
- Jose Leiva
- Yasemin Altun
- Bernhard Schölkopf

Funding by German Research Foundation, Max Planck Society & MSKCC

Thank you for your attention!

References

Caruana, R. (1997). Multitask learning. Machine Learning, 28(1):41–75.

Daume III, H. (2007). Frustratingly easy domain adaptation. In Conference of the Association for Computational Linguistics (ACL), Prague, Czech Republic.

Evgeniou, T., Micchelli, C., and Pontil, M. (2005). Learning multiple tasks with kernel methods. Journal of Machine Learning Research, 6(1):615–637.

Hsieh, C., Chang, K., Lin, C., Keerthi, S., and Sundararajan, S. (2008). A dual coordinate descent method for large-scale linear SVM. Proceedings of the 25th international conference on Machine learning, pages 408–415.

Kloft, M., Brefeld, U., Sonnenburg, S., Laskov, P., Müller, K.-R., and Zien, A. (2009). Efficient and accurate lp-norm multiple kernel learning. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C. K. I., and Culotta, A., editors, Advances in Neural Information Processing Systems 22, pages 997–1005. MIT Press.

Rätsch, G., Sonnenburg, S., and Schäfer, C. (2006a). Learning interpretable svms for biological sequence classification. BMC Bioinformatics, 7(Suppl 1):S9.

Rätsch, G., Sonnenburg, S., and Schäfer, C. (2006b). Learning interpretable svms for biological sequence classification. BMC Bioinformatics, 7 Suppl 1:S9.

Rätsch, G., Sonnenburg, S., Srinivasan, J., Witte, H., Müller, K.-R., Sommer, R., and Schl̈kopf, B. (2007). Improving the c. elegans genome annotation using machine learning. PLoS Computational Biology, 3(2):e20.

Rifkin, R. M. and Lippert, R. A. (2007). Value regularization and Fenchel duality. J. Mach. Learn. Res., 8:441–479.

Schweikert, G., Widmer, C., Schölkopf, B., and Rätsch, G. (2008a). An empirical analysis of domain adaptation algorithms. In Proc. NIPS 2008, Advances in Neural Information Processing Systems. accepted.

Schweikert, G., Widmer, C., Schölkopf, B., and Rätsch, G. (2008b). An empirical analysis of domain adaptation algorithms. In Advances in Neural Information Processing System, NIPS, volume 22, Vancouver, B.C.

Schweikert, G., Zien, A., Zeller, G., Behr, J., Dieterich, C., Ong, C. S., Philips, P., De Bona, F., Hartmann, L., Bohlen, A., Krüger, N., Sonnenburg, S., and Rätsch, G. (2009). mgene: Accurate svm-based gene finding with an application to nematode genomes. Genome Research. Advance access June 29, 2009.

References

Sonnenburg, S. and Franc, V. (2010). Coffin: A computational framework for linear SVMs. In Fürnkranz, J. and Joachims, T., editors, ICML, pages 999–1006. Omnipress.

Sonnenburg, S., Rätsch, G., Henschel, S., Widmer, C., Behr, J., Zien, A., de Bona, F., Binder, A., Gehl, C., and Franc, V. (2010). The SHOGUN Machine Learning Toolbox. Journal of Machine Learning Research, 11:1799–1802.

Sonnenburg, S., Rätsch, G., and Rieck, K. (2007). Large scale learning with string kernels. In Bottou, L., Chapelle, O., DeCoste, D., and Weston, J., editors, Large Scale Kernel Machines, pages 73–103. MIT Press, Cambridge, MA.

Sonnenburg, S., Zien, A., Philips, P., and Rätsch, G. (2008). Poims: positional oligomer importance matrices–understanding support vector machine-based signal detectors. Bioinformatics, 24(13):i6–14.

Toussaint, N., Widmer, C., Kohlbacher, O., and Rätsch, G. (2010). Exploiting physico-chemical properties in string kernels. BMC Bioinformatics, 11(Suppl. 8):S7.

Widmer, C., Görnitz, N., Zeller, G., and Rätsch, G. (2011). Hierarchical multitask structured output learning for large-scale sequence segmentation. submitted.

Widmer, C., Kloft, M., Görnitz, N., and Rätsch, G. (2012). Efficient training of graph-regularized multitask svms. In Proc. ECML.

Widmer, C., Leiva, J., Altun, Y., and Rätsch, G. (2010a). Leveraging sequence classification by taxonomy-based multitask learning. In Berger, B., editor, RECOMB, volume 6044 of Lecture Notes in Computer Science, pages 522–534. Springer.

Widmer, C., Leiva, J., Altun, Y., and Rtsch, G. (2010b). Leveraging sequence classification by taxonomy-based multitask learning. In Proc. RECOMB'10. accepted.

Widmer, C. and Rätsch, G. (2011). Transfer learning in computational biology. In Proc. ICML.

Widmer, C., Toussaint, N., Altun, Y., and Rätsch, G. (2010c). Inferring latent task structure for multi-task learning by multiple kernel learning. BMC Bioinformatics, 11(Suppl. 8):S5.

References

Zeller, G., Clark, R., Schneeberger, K., Bohlen, A., Weigel, D., and Ratsch, G. (2008). Detecting polymorphic regions in arabidopsis thaliana with resequencing microarrays. Genome Res, 18(6):918–929.

Zien, A., Krämer, N., Sonnenburg, S., and Rätsch, G. (2009). The feature importance ranking measure. In Buntine, W., Grobelnik, M., Mladenic, D., and Shawe-Taylor, J., editors, Proc. ECML PKDD, volume 5782/2009 of Lecture Notes in Artificial Intelligence, pages 694–709, Springer Berlin / Heidelberg. Springer.